WEST

Freeform Search

Database:
- US Patents Full-Text Database
- US Pre-Grant Publication Full-Text Database
- JPO Abstracts Database
- EPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Term: `L4 and (extract$ or view$)`

Display: 50 Documents in Display Format: FRO Starting with Number 1

Generate: ○ Hit List ● Hit Count ○ Side by Side ○ Image

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Monday, September 22, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L5	L4 and (extract$ or view$)	26	L5
L4	L3 and sentence$	26	L4
L3	L2 and similarity	42	L3
L2	(phrase$ near document$)	195	L2
L1	(document near summary)	308	L1

END OF SEARCH HISTORY

WEST

☑ Generate Collection Print

L5: Entry 13 of 26 File: USPT Mar 26, 2002

DOCUMENT-IDENTIFIER: US 6363377 B1
TITLE: Search data processor

Detailed Description Text (9):
While the exemplary embodiments of the invention are described as using a hashing function to cluster the query results, it is contemplated that other methods of clustering, may be used instead of the hashing function. One such alternate method might be to form a concordance. Clean phrases in each document may be alphabetically sorted as they are received to form a list of all of the words in the combined documents. Each item in the list may include the clean phrase, a list of the documents in which the clean phrase occurs and the offset in each document at which the clean phrase occurs. This concordance may be used to cluster clean phrases in the documents based on the occurrence of single words or on the near occurrences of groups of words in the documents. Another alternate method might be to form a vector for each document in the multidimensional space defined by all the clean phrases in the documents. Each dimension of this space can correspond to a single clean phrase in the document collection, and the corresponding position in a document's vector is set to 1 if the document contains the clean phrase and to 0 otherwise. Any number of geometric clustering algorithms can then be used to cluster the vectors into a small number of clusters so as to minimize a geometric measure of the cluster, such as the volume of the cluster or the cluster's diameter.

Detailed Description Text (11):
In the exemplary embodiment, the clustering algorithm is implemented in the language Perl, which includes a non-collision hashing function. An exemplary embodiment hashes each clean phrase from the document title, URL, and summary to any entry in the hash table (also known as a hash bucket) using the hashing function in Perl. The exemplary hash table entry includes counts of the number of documents that contain the hashed clean phrase. At the end of the hashing process, each entry in the table may or may not represent a cluster. The entries are analyzed to determine the best clusters by weighing both the number of documents that contain the common clean phrase and the length of the clean phrase. The best clusters are output to the user.

Detailed Description Text (13):
The user may choose to view any of the discovered clusters; the system, then, displays the documents that appear in the selected cluster. For example, as shown in FIG. 4, if the user were to choose cluster 410, the system would display the 23 documents that contain "www.quickaid.com/airports".

Detailed Description Text (14):
FIG. 5 shows an example of a graphical display of a search query for a second exemplary embodiment of the invention. The clustering lenses interface consists of a display of the title, URL, content and age lenses and a Combination window. For each lens, the corresponding part of each matching document is analyzed. As a result, a small number of interesting patterns are discovered and presented to the user by using pattern matching and clustering algorithms. Users also have the option of specifying their own patterns. More specifically, each tool takes one field at a time and partitions all the documents returned by the search engine according to a pattern found in that field. The documents may be partitioned into 1 to 5 clusters or more. Since the pattern analysis is performed on each field separately, it corresponds to viewing the documents through a lens that only displays the field of interest and hides the other fields.

Detailed Description Text (15):
FIG. 5 shows an illustration of a display for a query about New Jersey restaurants. For example, this query produces 100 matching documents. Title lens 500 partitions the documents found into 3 clusters corresponding to cells 502, 504 and 506. Title Lens 500

considers similarities in the titles of the matching documents. Searching for similarity in both format and words does the partitioning. For example, a format similarity is documents with "No Title" or documents whose title begins with "Re:". A word similarity refers to any common subsequence of words in the title. The strongest word similarity is identical titles; a weaker word similarity is an identical phrase within titles or identical words separated by other words, e.g. "Jane K. Doe' and Jane Katherine Doe".

Detailed Description Text (17):
Also shown in FIG. 5 is URL Lens 510 which partitions the 100 documents found into four clusters corresponding to cells 512, 514, 516 and 518. URL Lens 510 considers similarities in the matching documents' Web addresses. For example, if there are many files with "pub/biblio" as part of the pathname, they may form a cluster. In general, any nontrivial contiguous part of the file path is mined for patterns. URL lens 510 finds 40 URLs that contain the term "www.njweb.com/dining" corresponding to cell 512. In cell 514, URL lens 510 finds 20 URLs that contain the term "yahoo.com". In cell 516, URL lens 510 finds 20 URLs that contain the term "metrocast.com". In cell 518, URL lens 510 finds 20 URLs that have no patterns. Furthermore, the 40 URLs having "www.njweb.com/dining" as a substring are exactly those with titles "NJWeb: Dining in New Jersey". Such a fact is indicated by the edges 550 joining cells 502 and 512. Edges 552 indicate that the documents clustered in Cell 504 are exactly those documents clustered in cell 514.

Detailed Description Text (18):
Further, FIG. 5 shows Content Lens 520 with the 100 documents found partitioned into 4 clusters corresponding to cells 522, 524, 526 and 128. Content lens 520 considers similarity in the short excerpts of the matching documents. Content lens 520 first eliminates stop words, such as "a", "an", "the", "to", etc., and then tries to partition the documents by common sentences, phrases or words. As an example, content lens 520 finds 40 documents that contain the term "Home allendale bayonne belleville bergenfield bloomfield butler" corresponding to a cluster in cell 522. In cell 524, content lens 520 finds 20 documents that contain the term "top business and economy companies restaurants organizations". In cell 526, content lens 520 finds 20 documents that contain the term "cape may county". Contact lens 520 finds 20 documents that have no patterns in cell 528. Edges 554 indicate that documents clustered in cell 512 are exactly the same documents corresponding to cell 522. Edges 556 indicate that the documents found in cell 514 are exactly the same as the documents clustered in cell 524.

CLAIMS:

12. A system for organizing a search engine's results including a set of documents each document including a plurality of fields, comprising: a) means for receiving the set of documents; b) means for analyzing several fields from each document to determine patterns c) means for partitioning each document by the fields analyzed in step (b) into clusters based on a shared pattern; d) means for analyzing the partitions to identify clusters for display; e) means for displaying the clusters of documents to the user; and f) means for user selection of the clusters for viewing; wherein the means for analyzing the partitions to identify the clusters for display includes means for analyzing a best cluster for display by weighing both a number of documents that contain the shared pattern and a length of the shared pattern.

14. A carrier including a computer program which, when executed by a processor, causes the processor to organize a set of documents into clusters, by causing the computer to perform the steps of: a) receiving the set of documents; b) analyzing at least one field from each document to determine a pattern; c) partitioning each document by the fields analyzed in step (b) into clusters based on a shared pattern; d) analyzing the partitions to identify the clusters for display; e) displaying the clusters to the user; and f) allowing the user to select one or more of the clusters for viewing; wherein the step of analyzing the partitions to identify the clusters for display includes the step of analyzing a best cluster for display by weighing both a number of documents that contain the shared pattern and a length of the shared pattern.

WEST

☑ Generate Collection Print

L5: Entry 9 of 26 File: USPT Oct 22, 2002

DOCUMENT-IDENTIFIER: US 6470307 B1
TITLE: Method and apparatus for automatically identifying keywords within a document

Abstract Text (1):
A trainable method of extracting keywords of one or more words is disclosed. According to the method, every word within a document that is not a stop word is stemmed and evaluated and receives a score. The scoring is performed based on a plurality of parameters which are adjusted through training prior to use of the method for keyword extraction. Each word having a high score is then replaced by a word phrase that is delimited by punctuation or stop words. The word phrase is selected from word phrases having the stemmed word therein. Repeated keywords are removed. The keywords are expanded and capitalisation is determined. The resulting list forms extracted keywords.

Brief Summary Text (8):
3. Keywords are generated automatically by first automatically tagging the words in the document by their part-of-speech, such as noun, verb, adjective, etc., and then listing the most frequent noun phrases in the document.

Brief Summary Text (15):
In view of the limitations of the prior art methods of keyword generation, it is an object of this invention to provide a method and means for automatically generating keywords, that overcomes many of these limitations.

Brief Summary Text (21):
In accordance with the invention there is further provided a method of generating a plurality of keywords from an electronic, stored document including phrases, stop words delimiting the phrases, and punctuation. A computer is used to select from the document, raw phrases comprised of one or more contiguous words excluding stop words. A form of the raw phrases is used to generate the plurality of keywords in dependence upon a plurality of weighted criteria, wherein weights for the criteria are determined by a step of training. For example, the step of training is performed by providing a training document; providing a set of keywords that are dependent upon the training document; providing a set of weights that are independent of the training document; performing keyword extraction on the training document; comparing the generated keywords with the provided keywords; and then modifying the weightings for the criteria and repeating the step of training until the comparison is within predetermined limits. For example, training may be performed with a genetic algorithm and weights may be stored in a decision tree.

Brief Summary Text (22):
In accordance with the invention there is provided a method of generating a plurality of keywords from an electronic, stored document including phrases, stop words delimiting the phrases, and punctuation. A first list of words within the document that are not stop words are generated. Each word in the list is evaluated to determine a score in dependence upon a plurality of indicators and weights for each indicator, scores for different words in the list determined using same indicators and same weights. The list of words is ordered in dependence upon the scores. For each word in the list, all raw phrases of one or more words containing a word having a predetermined similarity are selected and a score for each selected word phrase is determined. Then the word in the list is replaced with a most desirable word phrase comprising a word having a predetermined similarity.

Drawing Description Text (3):
FIG. 3 is a simplified flow diagram of a method of extracting keywords form a text document according to the invention;

Drawing Description Text (4):
FIG. 4 is a simplified flow diagram of a method of training a keyword extraction system according to the invention; and,

Drawing Description Text (5):
FIG. 5 is a simplified flow diagram of a method of training a keyword extraction system using a genetic algorithm according to the invention.

Detailed Description Text (3):
Referring now to FIG. 2, a simplified flowchart is shown illustrating an embodiment of the invention. An initialisation step is performed wherein a document for analysis is input and stored in memory 110. The document is operated upon by an extractor, in the form of a plurality of procedures stored in memory comprising a plurality of computer coded instructions.

Detailed Description Text (4):
The extractor is provided with a text file of the document as input data and generates a list of keywords comprising words and phrases as output. The output keywords are intended to serve as a "short-summary" of the input text file or as a list of words and phrases for facilitating locating the document. Throughout this specification the term keyword refers to a keyword having one or more words and includes keyphrases.

Detailed Description Text (5):
In a preferred embodiment of this invention, the extractor has twelve parameters that determine how the input text from document is processed. These twelve parameters are determined and set using a standard machine learning paradigm of supervised learning. Referring to FIG. 4, a simplified flow diagram of a method of training a keyword extraction system according to the invention is shown. The method employs a genetic algorithm for this purpose and is described in more detail hereinbelow. The extractor is tuned with a data set comprising documents paired with target lists of keywords supplied by the author of the documents. The data set is easily assembled by referring to existing documents in a same field as that in which the method is to be employed, and selecting some documents and associated keywords. Since a human compiled list of keywords is generally the best, it is preferable to use such a list for training the system. Thus, the learning process involves adjusting the twelve parameters, described hereafter in greater detail, to maximise the match between the output of the algorithm and the target keyword lists--those keywords provided with the training data. The success of the learning process is measured in accordance with a match of generated keywords with the training data.

Detailed Description Text (6):
A description follows of how the twelve parameters are tuned, including a description of the core algorithm of the extractor and the functions of the parameters.

Detailed Description Text (7):
The twelve parameters in the extractor are tuned by the Genitor genetic algorithm (Whitley, D. (1989), The GENITOR algorithm and selective pressure, Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89), Morgan Kaufmann, pp. 16-121), to maximise performance using the training data. The performance measure--F-measure--is based on precision and recall: number of machine phrases=number of phrases output by the extractor number of target phrases=number of keywords associated with a same document from the training data set precision=number of matches between the generated keywords and those supplied with the training data set/number of machine phrases recall=number of matches between the generated keywords and the keywords supplied with the training data set/number of target phrases F-measure=(2*precision*recall) (precision+recall)

Detailed Description Text (8):
A phrase generated by the extractor is said to "match" a phrase in the target list when the two phrases contain the same sequence of stems. A "stem" is a word having no suffix or a word with its suffix removed. For the matching algorithm, preferably, a different stemming algorithm is used than for keyword generation.

Detailed Description Text (10):
Although either "evolutionary psychology" or "evolutionary psychologist" matches the target "evolutionary psychology"--they all correspond to the sequence of stems "evolut psycholog," this is counted as only one match. This prevents overvaluing extracted keywords when stemming fails to identify two words having a same stem.

Detailed Description Text (11):
Genitor is used to train the extractor by substantially optimising values of the 12 parameters; Genitor is not used during keyword generation as the training process is complete. When the optimal parameter values are known, Genitor need not be used. Referring again to FIG. 3, the method of extracting keywords is further explained below.

Detailed Description Text (13):
The Alogorithm The extractor executes the following steps. First, stems of single words, excluding stop words, are extracted and for each, a score is determined. The stems are ranked in accordance with their scores, from most desirable score to least desirable score. Stem phrases of one or more words are then extracted and scored in a similar fashion to the stems of single words. Additional parameters allow for emphasis on longer phrases or phrases of predetermined lengths. The stems of single words are "expanded" by replacing a stem of a single word with a highest scoring stem phrase comprising the stem. Duplicates are removed from the list and suffixes are added using a suffix adding procedure. A more detailed description of the procedures followed by the extractor follows.

Detailed Description Text (19):
4. FIND STEM PHRASES: A list is made of all phrases in the input text. A phrase is defined as a sequence of one or more words that appear consecutively in the text with no intervening stop words or sentence boundaries. Optionally, phrases are limited to less than a predetermined number of words. In the preferred embodiment, phrases are limited to three or fewer words. Characters in the phrases are all converted to lower case characters as necessary. Each phrase is stemmed by truncating each word in the phrase to at most STEM_LENGTH characters. Truncation of words within phrases has similar advantages to those set out with reference to truncation of single words. The stems of each word in a phrase are formed into stem phrases. For example, "Psychological Association decision" becomes a stem phrase of "psych assoc decis" when STEM_LENGTH is 5.

Detailed Description Text (33):
Referring to FIG. 4, a simplified flow diagram of a method of training a keyword extraction system according to the invention is shown. The method accepts a data set comprising text documents and keywords for the documents. Keywords are extracted according to the method to be trained and during training extracted keywords are evaluated against the provided keywords. Parameters used during keyword extraction are modified based on the differences between the sets of keywords. When the differences are less than a predetermined threshold, the training stops and the parameters determined through training are used for keyword extraction. It is of note that training is most effective when the system is trained for documents of a certain type or in a certain academic field. For example, providing the training system with ten documents on natural language generation using a computer will likely produce parameters most effective in extracting keywords from articles on natural language generation.

Detailed Description Text (34):
Referring to FIG. 5, a simplified flow diagram of a method of training a keyword extraction system according to the invention is shown. The method employs a genetic algorithm for this purpose. An initial population of parameter values are provided. As well, a training set comprising a sample training document and a list of author compiled keywords for that document is stored in memory. For each member of the population--set of parameter values--the method outlined with reference to FIG. 3 is executed and resulting keywords are analysed. The analysis results in a score for each extracted list of keywords. Preferably training is performed using a plurality of training documents, thereby resulting in a plurality of scores for each member of the population. Members in the population with least desirable scores are discarded. Least desirable scores are determined either as lowest average score or, alternatively, through a ranking system based on all scores.

Detailed Description Text (40):
Once an "ideal member" is selected, the parameters of the "ideal member" are used in the algorithm of FIG. 3. Training is complete and the algorithm is applied to documents for keyword extraction using those parameters. In practice, training is performed during installation and subsequent use of the method requires no further training. Of course, training is repeated when keyword extraction is not performed as well as desired.

Other Reference Publication (6):
Sonderland, S., Lehnert, W., "Wrap-Up: A Trainable Discourse Module for Information Extraction," Journal of Artificial Research, vol. 2, 1994, pp. 131-158, XP-002077091.

CLAIMS:

15. A method of generating a plurality of human intelligible keywords from an electronic, stored document including phrases, stop words delimiting the phrases, and punctuation, the method comprising the steps of: aa) providing a plurality of indicators and a weight associated with each of the indicators, each indicator and associated weight indicative of word/phrase significance within the document; a) generating a list of words within the document that are not stop words for determining a score in dependence upon an evaluation of each word of the list in dependence upon the plurality of indicators and the associated and same weights for each indicator, scores for different words in the list determined using same indicators and same weights; b) ordering the list of words in dependence upon scores; contiguous words excluding stop words: and, c) for each word in the list, selecting all raw phrases of one or more words containing a word having a predetermined similarity for determining a score for each selected raw phrase; and, d) replacing said word in the list with a most desirable word/phrase comprising a word having a predetermined similarity.

Set Name **side by side**	**Query**	**Hit Count**	**Set Name** **result set**
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L24	L23 and rule$	8	L24
L23	L22 and (tree near structure)	12	L23
L22	L21 and (match$ or sort$)	51	L22
L21	L20 and extract$	57	L21
L20	L19 and similarity	65	L20
L19	l13 and phrase$	218	L19
L18	L17 and similarity	5	L18
L17	L13 and (extract$ near phrase$)	14	L17
L16	L15 extract$	966162	L16
L15	L14 and similarity	24	L15
L14	L13 and phase$	142	L14
L13	(summary near document)	678	L13
L12	L4 and sentence$	0	L12
L11	L7 and sentence$	0	L11
L10	L9 and sentence$	0	L10
L9	L7 and extract$	21	L9
L8	L7 and (intersection near table)	0	L8
L7	L4 and summary	90	L7
L6	L4 and (document$ near summary)	0	L6
L5	L4 and (document$ near collection)	0	L5
L4	(phase near intersection)	140	L4
L3	L2 and (phase near intersection)	0	L3
L2	L1 and (document near retriev$)	704	L2
L1	search$ near retriev$	4688	L1

END OF SEARCH HISTORY

WEST

Freeform Search

Database:
- US Patents Full-Text Database
- US Pre-Grant Publication Full-Text Database
- JPO Abstracts Database
- EPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Term: L23 and rule$

Display: 50 **Documents in Display Format:** - **Starting with Number** 1

Generate: ○ Hit List ● Hit Count ○ Side by Side ○ Image

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Monday, September 22, 2003 Printable Copy Create Case

WEST

☑ Generate Collection | Print

L5: Entry 16 of 26 File: USPT Aug 1, 2000

DOCUMENT-IDENTIFIER: US 6098034 A
TITLE: Method for standardizing phrasing in a document

Abstract Text (1):

A method for standardizing phrases in a document includes the steps of identifying phrases of a document to create a preliminary list of standard phrases; filtering the preliminary list of standard phrases to create a final list of standard phrases; identifying candidate phrases of the document which are similar to the standard phrases; confirming whether a candidate phrase of the document is sufficiently proximate to the standard phrase to constitute an approximate phrase; and computing a phrase substitution to determine the appropriate conformation of standard phrase to the approximate phrase or the approximate phrase to the standard. Further this invention relates to a computer system for standardizing a document.

Brief Summary Text (2):

The problem addressed by this application is the identification in a document of user significant phrases, as indicated by repetition of particular sequences of words. Typically, in documents created by precision-oriented users of various sorts, certain groups of words are used to convey a particular idea. Each time the user desires to express that idea, the user prefers to utilize the same phrasing over others in order to avoid confusion of meaning. In order to determine whether a significant phrase has already been used in the documents, the user must review the document to extract the relevant sequence of words. Once a phrase has been extracted, the user may refer to it on a continual basis to ensure that, whenever that user desires to express a similar idea, a form of the extracted sequence of words is used.

Brief Summary Text (3):

Problems related to, and auxiliary to, the problem of extracting user-created standard phrases include: the identification of significant sequences of words nested within otherwise significant sequences of words; and establishing equivalence of nearly identical sequences of words where the only difference among the sequences relates to certain known, structural elements. Problems related to, and auxiliary to, the problem of extracting user-created phrases that are substantially similar to user-created standard phrases include: the computation of the phrase(s) that transform the substantially similar phrase into the user-created standard phrase or that transform the user-created standard phrase into the substantially similar phrase, standardizing the discrepancies of the two phrases while retaining the remainder of the attributes and content of the conformed phrase.

Brief Summary Text (5):

Further, the human reviewer seeks to identify similar yet non-identical phrases in order to conform them. There is generally no explicit extraction and designation of standard phrases; these phrases are left within their contexts and simply used as the standards to which similar

Brief Summary Text (6):

expressions must conform. Similarly, there is no explicit extraction and designation of phrases substantially similar to standard phrases. These phrases are also left within their contexts and are either conformed to the significant phrases to which they are substantially similar or are used as the master phrasing to which other similar phrases are standardized, including even the phrasing that constitutes the user standard phrasing.

Brief Summary Text (8):

In addition, the suffix tree is usually used for the applications of storage, compression, and searching. In the subject application, the tree is used not for document or phrase storage, but rather for phrase identification by establishing word sequences that satisfy the criteria for length and recurrence in the document. In more detail, each node of the tree is associated with a record of the number of occurrences of the word sequence at that node; any such word sequence of sufficient length, where the number of occurrences exceeds the required threshold, is preliminarily designated a phrase. Inclusion on the final phrase list follows the post-processing steps outlined below.

Brief Summary Text (10):
An algorithm for the construction of a word-based suffix tree has been published by Andersson, et al. (Andersson, A. Larsson Jesper N. Swanson, K. "Suffix Tree on Words," Department of Computer Science, Lund University, Jul. 12, 1995.) Andersson, et al. is neither related to nor contains aspects related to the subject invention because Andersson does not relate at all to the overall process that is the subject of this application, standardizing document phrasing. Further, Andersson deals only with the construction of a word-level suffix tree; it does not relate at all to the process of standard phrase extraction. Further, Andersson constructs its word-based suffix tree on the level of the entire document and does not innovate the sentence suffix tree structure that enables the subject method its unique combination of efficiency and non-complexity. Further, Andersson does not attempt to pre-process the text at all through stemming and abstraction of known characters. Lastly, Andersson, does not address the related problem of nested phrases or any resolution thereof.

Brief Summary Text (12):
The subject invention is a method for standardizing user phrasing in a user-created document. This method involves two separate components, each of which is further divided into separate steps. The first component involves the automatic extraction from the document of sequences of words constituting significant user phrases. This is accomplished through the combination of document pre-processing, the representation and analysis of the document text on a modified suffix tree, and heuristic post-processing. The second component involves the automatic extraction from the document of sequences of words that are significantly similar but not identical to significant user phrases, and the automatic generation of suggested phrasing for this approximately matched phrasing that conforms its phrasing to the standard. This is accomplished through a combination of the location of candidate word sequences, a computation of the weighted "edit distance" between the significant phrase and the phrase approximately similar to it, and certain natural language processing techniques. The overall result of this method is a list of significant user-created standard phrases and the standardization of approximately matched phrasing throughout the document.

Detailed Description Text (2):
The present invention relates to a method for standardizing phrases in a document, which comprises the steps of: identifying phrases of a document to create a preliminary list of standard phrases; filtering the preliminary list of standard phrases to create a final list of standard phrases; identifying candidate phrases of the document which are similar to the standard phrases; confirming whether a candidate phrase of the document is sufficiently proximate to the standard phrase to constitute an approximate phrase; and computing a phrase substitution to determine the appropriate conformation of standard phrase to the approximate phrase or the approximate phrase to the standard.

Detailed Description Text (3):
In one embodiment the step of identifying phrases of a document to create a preliminary list of standard phrases further comprises tokenizing the document. In another embodiment of the invention the step of identifying phrases of a document to create a preliminary list of standard phrases further comprises constructing sentence suffix trees for determination of standard phrases. In another embodiment of the invention the step of identifying phrases of a document to create a preliminary list of standard phrases further comprises traversing and analyzing the suffix tree. In another embodiment the step of identifying phrases of a document to create a preliminary list of standard phrases further comprises the application of a stop list.

Detailed Description Text (4):
The high level design of the first component of the process, the automatic extraction of standard, user phrases, is centered around the construction of a modified suffix tree in which stemmed words are the atomic units; this tree is traversed and analyzed to identify sequences of words meeting certain pre-set criteria for significant phrases. The use of the suffix tree is supplemented by several additional steps that, together, enhance the accuracy and efficiency of the process, and hence the number of

standard phrases found. First, the processes of stemming, application of a stop list, and abstraction for known structural elements integrate information retrieval and other techniques to ignore elements of the document that do not affect the overall meaning of the sequence of words. These processes thus broaden the candidate group of phrases. Second, effectiveness is further enhanced by the post-construction processes of eliminating nested sub-phrases that do not occur sufficiently frequently independent of their nesting phrases. Finally, validity of the extracted phrase is supplemented by eliminating dangling minimal content words.

Detailed Description Text (5):
As a result of these additional processes, the extraction of standard phrases by the process is not compromised by differences in pluralization, capitalization and conjugation nor by the use of differing minimal content words, nor is it affected by variations in usage of certain structural elements.

Detailed Description Text (10):
Following the analysis of the suffix tree to extract phrase candidates, certain additional processing is added to heighten the accuracy of the phrase finding. First, the phrases identified on the suffix tree are sorted and reviewed in order to determine whether any phrase is wholly nested within any other phrase. Such nested phrases may be standard phrases in their own right, but in order to be designated as such, they must be of sufficient length and must occur outside the nesting phrase a certain number of times. Further, phrases may be multiply nested within an overall nesting phrase. As a result, where there is one sequence of words wholly nested within a larger sequence and the nested sequence does not meet the criteria for length and recurrence, it will not be considered a significant phrase; however, a third sequence of words, wholly nested within the first two, may still be designated a standard phrase if it meets these criteria.

Detailed Description Text (12):
The high level design of the second component of the process, the extraction of user-created phrases that are substantially similar to standard user-created phrases, is centered around a calculation of minimal "edit distance." The use of the edit distance calculation is supplemented by a pre-processing step that greatly enhances the efficiency of the method.

Detailed Description Text (13):
In one embodiment the step of identifying candidate phrases of the document which are similar to the standard phrases of the above invention the standardizing method further comprises constructing of a dictionary and phrase index. In another embodiment of the subjetc invention the step of identifying candidate phrases of the document which are similar to the standard phrases further comprises identifying candidate phrases by searching for approximate matches. In another embodiment of the subject invention the step of identifying candidate phrases of the document which are similar to the standard phrases further comprises application of a shifting window. In another embodiment of the subject invention the step of identifying candidate phrases of the document which are similar to the standard phrases further comprises accumulation of counts. In another embodiment of the subject invention the step of identifying candidate phrases of the document which are similar to the standard phrases further comprises generating candidates.

Detailed Description Text (14):
The dictionary of standard phrases that forms the basis of the determination of substantial similarity, termed the "Dictionary," is broken down into two coordinating data structures. These structures include both a list of phrases and a character-based "Phrase Index" containing all the characters, stemmed words and locations of these stemmed words contained in the list of phrases. This dual-structured dictionary enables efficiency in comparing the text under analysis to the standard phrases and, in particular, filtration of the document in order to locate candidate word sequences that may be substantially similar to a user-created standard phrase. These candidate phrases are not necessarily substantially similar to the standard; they simply meet certain word usage criteria that make them worthy of further analysis via the edit distance calculation. As the edit distance computation is time-consuming, this filtration phase is necessary in order to markedly reduce the number of phrases for which edit distance is to be calculated. This finding of candidate phrases is accomplished through the traversal of the text by a "sliding window," seeking units of text containing certain words in common with the words contained in the dictionary of standard phrases. Where a sufficiently large number of words is found in common, irrespective of the order of their appearance, the sequence of words is designated a candidate phrase, deserving of

further analysis by the time-intensive edit distance calculation.

Detailed Description Text (15):
Confirmation of the similarity of candidate phrases is accomplished by an edit distance calculation which is a method for quickly calculating the number of operations necessary to perform on object 1 in order to transform it into object 2. The operations computed to determine the edit distance include insertion, deletion and replacement. The calculation enables identification of the item most closely matching the object or objects under analysis.

Detailed Description Text (20):
In one embodiment of the step of confirming whether a candidate phrase of the document is sufficiently proximate to the standard phrase to constitute an approximate phrase the standardizing method further comprises calculating the edit distance of candidate phrases. In another embodiment of the invention the standardizing method further comprises calculating the edit distance of candidate phrases. In another embodiment of the invention the standardizing method further comprises substituting conforming phrases. In another embodiment of the invention the standardizing method further comprises weighting the edit distance.

Detailed Description Text (25):
Further, "encoding" is defined as the process of transforming a sequence of words into a sequence of numbers, where each number represents a particular word (the word's code). "Lexical attribute" is defined as an attribute indicating if a particular word has independent meaning, or if its meaning is only in its grammatical function. "Master phrase" is defined as a word sequence constituting a standardized phrasing for a document. "Nested phrase" is defined as a pair of phrases P and P', such that the sequence of words in P' occurs as a subsequence of the sequence of words in P. P' is nested in P. Trie"is defined as a tree, each of whose nodes is labeled by a sequence of elements (e.g., characters or words), where each node represents the string formed by concatenating the labels of its ancestors starting with the root and ending with the label of the node itself". "Path compressed trie" is defined as a trie where no node has exactly one child. "Phrase" is defined as a sequence of words in canonical form constituting phrasing. "Prefix nested phrase" is defined as a nested phrase where the shorter phrase's word sequence is a prefix of the longer phrase's word sequence. "Segmentation" is defined as the process of converting a document represented as a sequence of characters into a sequence of sentences, each comprising a sequence of words.

Detailed Description Text (26):
Further, "semantic attribute" is defined as an attribute categorizing a sequence of words according to its meaning. For example, "Ab Def Inc." may be categorized as a "company name". Semantic categories may be organized in a hierarchy, with higher-level categories including many lower-level categories; a word is considered to have all semantic attributes whose categories it is included in. Note that in contradistinction to lexical attributes, semantic attributes may apply to sequences of several words. "Sentence suffix tree" is defined as a trie representing the set of sequences comprising all of the Suffixes of all of the sentences in a given document. "Structural element" is defined as a word, or sequence of words, in a document that serve a unified functional purpose in the document, independent of the words' individual meanings. In a particular domain, different types of structural elements will exist. For example, in the domain of legal documents, one type of structural element is the "defined term", corresponding to usage of a term defined elsewhere in the document. Identifying structural elements and their types in a document may be done either manually or automatically, through the use of domain-dependent methods. "Stemming" is defined as the process of reducing a word to its canonical form. "Stop list" is defined as a list of frequently used words which bear minimal content on their own (such as conjunctions or prepositions). "Suffix tree" is defined as path compressed trie which represents exactly the set of sequences comprising the distinct suffixes of a given sequence of elements.

Detailed Description Text (27):
Further, "syntactic attribute" is defined as an attribute giving a word's syntactic function. For example, in English text, this may be as simple as a part-of-speech label ("noun", "verb", etc.) or a more complex descriptor based on the word's relationship to other surrounding words. "Template replacement" is defined as a method of the automatic conforming, where for a certain type of words, the approximate phrase's word is retained and not replaced by the corresponding word in a master phrase (for example conjunctions--"and" cannot be replaced by "or"). "Tokenization" is defined as the

process of converting a document into a sequence of sentences, including the processes of segmentation, abstraction, stemming, and encoding. "Weighted edit distance" is defined as a variety of edit distance where the operations of insertion, deletion, and replacement are weighted differently, possibly according to the attributes of the words being compared.

Detailed Description Text (34):
The Candidate Phrase Finding Phase 6 assembles a preliminary group of word sequences that may be sufficiently similar to the phrases contained in the phrase dictionary (including those phrases extracted, as above, as well as any additional manually input phrases) to constitute candidate phrases 9 that the user may desire to conform to the standard phrase. Finally, the Candidate Phrase Confirmation Phase 8 quickly performs a weighted edit distance calculation on the candidate phrases to determine whether these phrases are sufficiently proximate to the standard phrases to constitute approximately matching phrases 11. If the phrases are determined to match sufficiently well, the Phrase Conformation Phase 10 then utilizes the results of the edit distance calculation in order to compute the proper form to substitute in for either the Master Phrase or the Approximate Phrase. This form attempts to retain as much as possible of the substituted-for phrase's grammatical structure. Once an approximation to the correct substitution form is computed, the user is presented with a candidate phrase for substitution that conforms the discrepancies between the standard phrase and the approximately matching phrase, retaining the syntactic coherence of the conformed phrase.

Detailed Description Text (35):
In more detail, the Master Phrase Finding Phase 2, detailed in FIG. 2 and below, itself consists of several steps. First, the pre-processing step involves the tokenization 12 of the document into sentence-based "word packs", the abstraction of known structural elements within the document, and the stemming of word packs. Second, the determination of phrases involves the construction of the suffix tree, the traversal and analysis of this tree, and the application of a "stop list" of words during this analysis. The output of this Phase is a preliminary list of user-specific phrases.

Detailed Description Text (37):
The Candidate Phrase Finding Phase 6, detailed in FIG. 3, and below, consists of several steps that, in the aggregate, limit the number of candidate word sequences by eliminating non-candidate word sequences. This Phase involves the construction of a dual-structured phrase dictionary, including a phrase list and a phrase index. The words contained in the phrase index are continually compared to the text contained in a "sliding window" to determine whether and where there is sufficient similarity between the text contained in the "window" and the phrases contained in the phrase dictionary. The output of this phase is a preliminary list of phrases sufficiently similar to the user-created standard phrases to merit further analysis via the next phase.

Detailed Description Text (46):
source sequence is divided into a set of "sentences", a "sentence suffix tree" is a trie whose set of sequences represented are all of the suffixes of each sentence in the sequence.

Detailed Description Text (49):
As depicted in FIG. 2, the Master Phrase Finding Phase 2 consists of three major steps, each detailed in this section. These include (a) tokenization, detailed in FIG. 6, (comprising segmentation, abstraction, stemming, and encoding), (b) construction of the Sentence Suffix Tree 15 (the "SST"), and (c) traversal and analysis of the SST, including application of the stop list. These steps together result in the construction of the candidate library of user phrases.

Detailed Description Text (52):
Segmentation of the source text involves its deconstruction into units, termed "word packs", each one representing a word. Segmentation is performed on a sentence-by-sentence basis. Upon reaching the end of a sentence, the word packs from that sentence are ready for the next stage of tokenization. The process breaks up the source text into both words and sentences in parallel, extracting words one by one until a sentence boundary is reached. A standard finite-state machine technique is used to recognize patterns that indicate word and sentence boundaries. This method is efficient, extensible, and simple.

Detailed Description Text (53):
As an example of segmentation, consider the following (schematic) sample sentence, in

which each of the letters below represents a word in that sentence. Further, assume that the word represented by the letter "a" below has different conjugated instances, as indicated below:

Detailed Description Text (55):
Segmentation divides the input characters in the source text into the sequence of words constituting this sentence, as a sequence of its component words. Once the complete sentence is produced, it is abstracted, as described below.

Detailed Description Text (58):
For example, assume the group of word packs tokenized from the above-provided sample sentence:

Detailed Description Text (70):
Once the sentence has been tokenized, abstracted, and stemmed, each word is encoded as a unique integer. The encoding is done in order to reduce the cost of comparing words to one another in the suffix tree construction and analysis phases, as well as during phrase filtering. Encoding is accomplished by representing each word in a trie, as described in Section IV(2), above. Each word is associated with a unique integer. When a new word is encountered, if it is found to appear already in the trie, the integer associated with that word is used as its code. Otherwise, the word is inserted into the trie and is assigned a new code. An ancillary array is also maintained, which indexes nodes in the trie corresponding to words according to their assigned codes. The trie together with this "code index" is termed the "Word Index". Special markers, such as those associated with structural element abstractions, have predefined integer codes that are inserted directly. The sequence of words is thus converted to a sequence of integers, enabling words to be most efficiently compared to one another.

Detailed Description Text (71):
The set of word packs has now been completely pre-processed and is ready to be represented by, and analyzed in, the Sentence Suffix Tree.

Detailed Description Text (73):
A. Construction of the Sentence Suffix Tree

Detailed Description Text (74):
The sentence suffix tree utilized as a part of the overall process described herein is a specialized form of suffix tree, as noted in Section IV(1), above. In essence, a suffix tree for a given element is a method of representing every distinct sub-sequence of items constituting a suffix of that element. Were the general form of suffix tree utilized in the context of a document, the suffixes that the suffix tree would represent would be those of the entire document. In this context, the atomic units represented at each node on the tree would be stemmed word packs, not characters or character strings. Each leaf of the resultant suffix tree would represent a distinct suffix of the document as a whole. The use of the SST, as further detailed below, yields a different and more efficient result.

Detailed Description Text (85):
This unique combination is enabled by the specific nature of the elements under analysis--documents--and the atomic units chosen--words. The nature of the document permits a significant and innovative modification in the method described herein, termed the EE Method. The significant phrases sought for identification in a document cannot by definition span a sentence boundary; if a whole sentence is a standard phrasing, then that sentence becomes designated a phrase. As a result, the Sentence Suffix Tree, constructed as part of this application, represents not the distinct sub-sequences of words in the document as a whole, but rather only the distinct sub-sequences of words in each individual sentence. This markedly reduces the time for construction of this tree to:

Detailed Description Text (87):
s=the number of words in a given sentence in the document

Detailed Description Text (88):
f=the number of sentences in the document

Detailed Description Text (89):
To simplify the equation, in the worst case scenario in a document with X sentences of length less than or equal to s.sub.max, the maximum time (T) that it takes to build a suffix tree is represented by the following formula:

Detailed Description Text (95):
X=1000 The number of sentences in a document;

Detailed Description Text (96):
s.sub.max =10 The number of words in each sentence in the document.

Detailed Description Text (101):
The EE Method is sufficiently efficient to allow construction of a Sentence Suffix Tree in near-linear time, yet accomplishes this with the far lesser complexity associated with the Q Method. This optimal combination of efficiency and complexity is highlighted as increasingly large documents are examined. It is important to recognize that, as documents grow, their number of sentences (A) grows, but the length of a sentence remains constant (s.sub.max). Therefore, as the only quadratic function in the EE Method is sentence length, which does not change with the length of the document, the time in the EE Method increases only linearly as the document grows in size.

Detailed Description Text (103):
As noted above, the SST constructed using the EE Method represents every distinct suffix subsequence of words. Each word sequence (suffix of a sentence) is inserted individually into the tree as follows (see the detailed flow-chart in FIG. 5). First, the tree is traversed from the root down in order to find the node with the longest corresponding word sequence which is a prefix of the current sequence (that to be inserted). If the node's sequence completes a prefix of the current sequence, a new child is added to that node, which child is labeled with the remaining suffix of the current sequence. Otherwise, the sequence corresponding to the node contains words beyond the corresponding prefix, and hence the node N must be split. This is done by creating a new node M, labeled with the first part of N's label (that contained in the prefix of the current sequence). N then becomes a child of M, and is labeled with the part of its old label remaining after M's label is removed. Then, the remaining suffix of the current sequence is inserted as a child of M, just as in the case where no splitting was required.

Detailed Description Text (107):
The tree is constructed iteratively, starting from the first term in the sentence and moving forward. Considering the suffix of the first term, the tree would appear as follows:

Detailed Description Text (113):
The tree at this point in the sentence would appear as follows:

Detailed Description Text (116):
Each node in this final sample tree represents distinct sub-sequences of words within the sentence, while each leaf represents a distinct suffix of a sentence. Beside each node is a number that represents the number of recurrences R of the particular word or set of words. The number of leaves is, at most, the number of words in the sentence. As a result, the maximal size of the tree is proportional to the size of the text as a whole.

Detailed Description Text (117):
As increasing numbers of sentences are examined by the EE Method, only incremental suffixes are added to the tree. Word sets and suffixes already contained on the tree are not added. Instead, a counter of the number of recurrences, R, is incremented.

Detailed Description Text (119):
As mentioned above, the SST constructed by the EE Method is used not for storage and compression, but rather for traversal and analysis, leading to the extraction of significant phrases. Analysis is based on two criteria that exist for the preliminary determination of qualification as a significant phrase, length of the phrase, L, and the amount of recurrence of the phrase, R. Only where both the criteria are met, i.e., where:

Detailed Description Text (133):
Phrases included on the preliminary list of phrases noted above are filtered in order to eliminate repetitiveness among the phrases and, further, to improve the natural language veracity of the preliminary phrase list. First, the elimination of repetitiveness is necessary, as the preliminary phrase list may include certain phrases that contain, wholly within them, nested phrases that themselves are evaluated to be significant phrases. If these nested phrases always appear only as a part of the

larger, nesting phrase, they should not be listed as separate entries on the final phrase list as distinct phrases. However, if they appear sufficiently on their own, independently of the larger nesting phrase, they should be included as significant phrases in their own right. Second, the phrases extracted in the traversal and analysis step described above may start or end with certain minimal content words that should not be considered parts of the phrase. To deal with this problem, certain dangling minimal content words are not included as part of the phrases contained on the final phrase list.

Detailed Description Text (228):
Clearly, the size of the window is an important factor in determining the number of words in any window that are potentially similar to the Master Phrases. The larger the window, the more words contained in it and therefore the larger the potential for similarity to the Phrase Index. In order to avoid both being so small as to be under-inclusive or being so large as to be over-inclusive, the size at which the window is set is to be slightly larger than the length of the longest phrase. At this size, the finding of a candidate phrase similar to even the longest Master Phrase is facilitated. The window is sized even larger than the longest Master Phrase in order to account for possible insertions.

Detailed Description Text (232):
This point is clarified by return to the example above. As the window slides forward from t=0 to t=1, the word w.sub.6 is added to the analysis, while the word w.sub.1 is removed from consideration. Only the changes that relate to these two words must be considered in determining the potential candidacy of the windowed text as an Approximate Phrase. The remainder of the words in the window remain the same. If w.sub.1 appeared once in the Phrase Index, upon the shifting of the window forward, the count of the number of words appearing in the Phrase Index must be decremented one. If w.sub.6 also appears in the Phrase Index, the net affect of the shift of the window is 0. However, if w.sub.6 does not appear in the Phrase Index, the net affect of the shift is the reduction by 1 of similarity to the Master Phrases. Similarly, if w.sub.1 does not appear in the Phrase Index, the affect of the shift of the window is either 0, if w.sub.6 also does not appear in the Phrase Index, or 1, if w.sub.6 does appear.

Detailed Description Text (242):
Calculation of edit distance is the subject of much prior art. The particular method described in this application is based on that of Ukkonen, although other efficient methods exist. Among other things, this calculation is heavily utilized in character-based error finding, as in spelling checkers. This calculation enables rapid calculation of similarity between items and is typically used on the level of character strings, not words.

Detailed Description Text (245):
There are three central aspects of the subject method for edit distance computation. First, edit distance is calculated on the level of the word in order to determine the similarity of one word sequence to another, rather than on the level of the character to determine the similarity of one character sequence (i.e., a word) to another. Furthermore, comparison is carried out on the canonical forms of words, in order to properly account for morphological variation (such as pluralization). As noted above in Section II, this word based analysis is most appropriate both because the natural atomic units of phrases are words and because word-based analysis can be considerably faster than character-based analysis.

Detailed Description Text (247):
Second, edit distance is not used in the subject method to recommend replacement of the object under analysis with a particular objectively correct item, e.g., the correct spelling of a word. In the context of phrasing there is not necessarily an objectively correct phrase. In theory, any phrasing might be correct and it is the user's preference that dictates the style chosen. As a result, the process of extracting approximately-similar phrases is oriented toward presenting the user with user-defined phrasing options for similar phrases, in which the user may opt to modify either phrase to suit the other, whether the Approximate Phrase to suit the Master, or the Master Phrase to suit the Approximate. Regardless of the option chosen, a similar amount of the original Phrase is retained, in order to preserve the syntactic coherence of the document.

Detailed Description Text (297):
The calculation of minimal edit distance is best demonstrated through the use of an example; the simple sample phrases below and the array at Table 1 are used to

demonstrate this point. Assume that P.sub.M is a Master Phrase which P.sub.C, the candidate phrase, may approximate. The Edit Distance Filter will determine whether the similarity between the two phrases is sufficient for the candidate phrase to be designated an Approximate Phrase.

CLAIMS:

1. A method of extracting phrases in a document, which comprise the steps of:

extracting phrases of a document to automatically create a preliminary list of extracted phrases;

filtering the preliminary list of extracted phrases to create a final list of extracted phrases;

extracting candidate phrases of the document which are similar to extracting phrases contained in the final list of extracted phrases;

confirming whether a candidate phrase of the document is sufficiently proximate to the extracted phrase to constitute an approximate phrase by calculating an edit distance of the candidate phrases based on two distinct cost functions, a first one relating to a semantic significance and role of a text of the document, and a second one elating to operations performed on the text of the document; and

computing a phrase substitution to determine the appropriate conformation of one of the extracted phrase to the approximate phrase and the approximate phrase to the extracted phrase.

2. The method of extracting phrases in a document of claim 1, wherein the step of extracting phrases of a document further comprises tokenizing the document.

3. The method of extracting phrases in a document of claim 1, wherein the step of extracting phrases of a document further comprises constructing sentence suffix trees for determination of extracted phrases.

4. The method of extracting phrases in a document of claim 3, wherein the step of extracting phrases of a document further comprises traversing and analyzing each of said suffix trees.

5. The method of extracting phrases in a document of claim 1, wherein the step of extracting phrases of a document further comprises applying a stop list.

6. The method of extracting phrases in a document of claim 1, wherein the step of filtering the preliminary list of extracted phrases further comprises extracting prefix nested phrases.

7. The method of extracting phrases in a document of claim 1, wherein the step of filtering the preliminary list of extracted phrases further comprises extracting suffix nested phrases.

8. The method of extracting phrases in a document of claim 1, wherein the step of filtering the preliminary list of extracted phrases further comprises eliminating duplicative nested phrases from the final list of induced phrases.

9. The method of extracting phrases in a document of claim 1, wherein the step of filtering the preliminary list of extracted phrases further comprises post-processing of the extracted phrase.

10. The method of extracting phrases in a document of claim 1, wherein the step of filtering the preliminary list of extracted phrases further comprises eliminating dangling words.

11. The method of extracting phrases in a document of claim 1, wherein the step of extracting candidate phrases further comprises constructing a dictionary.

12. The method of extracting phrases in a document of claim 1, wherein the step of extracting candidate phrases further comprises constructing a phrase index.

13. The method of extracting phrases in a document of claim 1, wherein the step of

extracting candidate phrases further comprises extracting candidate phrases by searching for approximate extracted matches.

14. The method of extracting phrases in a document of claim 1, wherein the step of extracting candidate phrases further comprises applying a shifting window of variable starting point, ending point, and size regardless of starting words of the candidate phase.

15. The method of extracting phrases in a document of claim 1, wherein the step of extracting candidate phrases further comprises accumulating counts.

16. The method of extracting phrases in a document of claim 1, wherein the step of extracting candidate phrases further comprises generating candidates.

17. The method of extracting phrases in a document of claim 1, wherein the step of confirming whether a candidate phrase of the document is sufficiently proximate to the extracted phrase further comprises computing a phrase conforming the approximate extracted phrase to the extracted phrase.

18. The method of extracting phrases in a document of claim 1, wherein the step of confirming whether a candidate phrase of the document is sufficiently proximate to the extracted phrase further comprises computing a phrase conforming the extracted phrase to the approximate extracted phrase.

19. The method of extracting phrases in a document of claim 1, wherein the step of confirming whether a candidate phrase is sufficiently proximate to the extracted phrase further comprises weighting an edit distance.

20. The method of extracting phrases in a document of claim 1, wherein the step of extracting phrases of a document is performed without using a pre-stored dictionary.

WEST

End of Result Set

☑ Generate Collection Print

L24: Entry 8 of 8 File: USPT May 25, 1999

DOCUMENT-IDENTIFIER: US 5907841 A
TITLE: Document detection system with improved document detection efficiency

Abstract Text (1):
A document detection system capable of detecting a desired document from a large number of documents easily and accurately in which the user can make a judgement concerning the appropriateness of the detection result quickly. In the system, those documents which contain a semantic structure of a detection command containing natural language expressions entered by a user are detected. Also, the keywords of each document can be extracted from the summary of each document and those documents whose keywords match with detection keywords specified by a user can be detected. Also, the summary of each detected document can be automatically generated according to text structures of each detected document and displayed along with the detected document itself. Also, the detection processing can be carried out with respect to the summaries of the documents instead of the documents themselves.

Brief Summary Text (13):
Moreover, in the conventional document detection system, in displaying the titles of the detected documents, the titles are simply arranged in a prescribed order according to the user's query such as an order of descending similarities to the keywords used in the detection key. For this reason, it has been impossible for the user to comprehend the relative relationships among the detected documents and the level of similarity with respect to the detection command for each of the detected documents from the displayed detection result, and consequently it has been difficult for the user to have an immediate impression for the appropriateness of the displayed detection result.

Brief Summary Text (14):
Furthermore, in the conventional document detection system, the detection scheme is limited to that in which each document as a whole is treated as a single entity, so that the document containing the desired content in the background section and the document containing the desired content in the conclusion section will be detected together in mixture. In other words, the detection result contains a variety of documents mixed regardless of viewpoints in which the desired contents appear in the documents. For example, if there is no interest in what had been done in the past, the detected document which matches with the given keywords in the background section will be of no use. Yet, in the conventional document detection system, the documents having different perspectives such as the document containing the desired content in the background section and the document containing the desired content in the conclusion section will not be distinguished, and the mixed presence of these documents in different perspectives makes it extremely difficult for the user to judge the appropriateness of the detection result.

Brief Summary Text (16):
Also, there has been a proposition for a scheme to reduce the burden on the user to read the entire content of each detected document by providing a man-made document summary for each stored document in advance in correspondence to each stored document itself and displaying the document summary at a time of displaying the detection result. However, in such a scheme, an enormous amount of human effort is required for preparing the document summary for each document at a time of producing the database itself, which is not practically justifiable unless the database system has a remarkably high utilization rate. Moreover, there are many already existing database systems in which the document summary for each document is not provided, and an enormous amount of human efforts is similarly required for preparing the document summary for each document in such an already existing database system. In addition, the

man-made document summary is produced in the very general viewpoint alone, so that there is no guarantee that each document is summarized from a viewpoint suitable for the required detection. As a result, the document summary displayed as the detection result can be quite out of point from the viewpoint of the user with the specific document detection objective, and in such a case, it is possible for the user to overlook the actually necessary document at a time of judging whether each detected document is the desired document or not.

Brief Summary Text (19):
It is another object of the present invention to provide a document detection system capable of automatically preparing and displaying a document summary for each document in a viewpoint which is efficiently comprehensible for the user, considering the limited visual data processing ability of the human user, such that the user can make a judgement concerning the appropriateness of the detection result quickly.

Brief Summary Text (21):
According to another aspect of the present invention there is provided a document detection system, comprising: input means for entering a detection command containing detection keywords specified by a user; document storage means for storing a plurality of detection target documents; summary generation means for generating a summary of each detection target document stored in the document storage means, and extracting keywords of each detection target document from the summary of each detection target document; and detection processing means for detecting those detection target documents stored in the document storage means whose keywords extracted by the summary generation means match with the detection keywords in the detection command entered at the input means as detected documents.

Brief Summary Text (22):
According to another aspect of the present invention there is provided a document detection system, comprising: input means for entering a detection command specified by a user; document storage means for storing a plurality of detection target documents; detection processing means for detecting those detection target documents stored in the document storage means which match with the detection command entered at the input means as detected documents; summary generation means for automatically generating a summary of each detected document obtained by the detection processing means according to text structures of each detected document; and detection result output means for displaying each summary generated by the summary generation means.

Brief Summary Text (23):
According to another aspect of the present invention there is provided a document detection system, comprising: input means for entering a detection command specified by a user; document storage means for storing a plurality of detection target documents; summary generation means for generating summaries of the detected documents stored in the document storage means; summary storage means for storing the summaries generated by the summary generation means; and detection processing means for detecting those summaries stored in the summary storage means which match with the detection command entered at the input means.

Drawing Description Text (9):
FIG. 8 is a diagrammatic illustration of an exemplary semantic analysis rule used in the flow chart of FIG. 6.

Drawing Description Text (10):
FIG. 9 is a diagrammatic illustration of an exemplary unnecessary expression rule used in the flow chart of FIG. 6.

Drawing Description Text (15):
FIGS. 14A and 14B are a flow chart for an operation of a keyword index matching unit and a document file set calculation unit in the detection processing unit of FIG. 10.

Drawing Description Text (17):
FIG. 16 is a flow chart for an operation of a semantic structure index matching unit in the detection processing unit of FIG. 10.

Drawing Description Text (25):
FIG. 24 is a table representing an exemplary display priority rule dictionary in the detection control unit shown in FIG. 2.

Drawing Description Text (38):

FIG. 37 is a diagrammatic illustration of an exemplary bibliographical matter analysis rules used in the fourth variation of the first embodiment.

Drawing Description Text (39):
FIG. 38 is a diagrammatic illustration of an exemplary unnecessary expression rule used in the fourth variation of the first embodiment.

Drawing Description Text (44):
FIG. 43 is a diagrammatic illustration of an exemplary keyword extraction rule dictionary used in the seventh variation of the first embodiment.

Drawing Description Text (87):
FIG. 86 is a flow chart for a matching relation extraction processing in the operation of FIG. 85.

Detailed Description Text (6):
In further detail, a main portion of the document detection system of this first embodiment has a functional configuration as shown in FIG. 2, which comprises: an input unit 11 for entering an input sentence in a natural language from the user; an input analysis unit 12 for carrying out various analyses of the input sentence including the morphological analysis, the syntactic analysis, and the semantic analysis; a detection processing unit 13 for carrying out the detection processing for detecting documents according to a syntactic analysis result obtained from the input sentence and a detection key constructed by using keywords extracted from the input sentence; a summary generation unit 14 for producing a summary of each detected document; a document storage unit 15, connected with the detection processing unit 13, for storing the document database; a detection result output unit 17 for outputting the results obtained by the input analysis unit 12, the detection processing unit 13, and the summary generation unit 14; an individual data storage unit 16, connected with the input analysis unit 12, the detection processing unit 13, the summary generation unit 14, and the detection result output unit 17, for storing individual data including the detected documents; and a detection control unit 18 for controlling the operations of the other processing modules including the input unit 11, the input analysis unit 12, the detection processing unit 13, the summary generation unit 14, and the detection result output unit 17, while managing user interactions.

Detailed Description Text (13):
In a case of the affirmative, next at the step 305, the syntactic and semantic analyses results for the detected documents prepared in advance and stored in the document storage unit 15 are matched with the syntactic and semantic analyses results for the input character string stored in the individual data storage unit 16. A set of documents detected and matched at this step 305 are temporarily stored in the individual data storage unit 16. Otherwise, the step 305 is skipped.

Detailed Description Text (14):
Next, at the step 306, the text contents of the detected and matched documents stored in the individual data storage unit 16 are taken out from the document storage unit 15, and the summary generation unit 14 is activated to produce a summary for each detected and matched document from the text content. The summary for each detected and matched document obtained by the summary generation unit 14 is then stored in the individual data storage unit 16.

Detailed Description Text (15):
Then, at the step 307, the summary generation unit 14 checks the matching of the keywords or the syntactic and semantic analyses results for the input sentence in the summary for each detected and matched document. The check result obtained at this step 307 is then stored in the individual data storage unit 16.

Detailed Description Text (16):
Finally, at the step 308, the detection result output unit 17 is activated to display the detection result containing the document names or the summaries of those documents remaining in the individual data storage unit 16 after the step 307, in an order according to the check result obtained at the step 307. Here, the detection result output unit 17 can display and change the data stored in the individual data storage unit 16 in accordance with the commands entered from the user through the input unit 11, such that the user can select the desired document from the displayed detection result.

Detailed Description Text (20):

Next, the input analysis unit 12 has a detailed functional configuration as shown in FIG. 5, which comprises: a morphological analysis unit 120, a syntactic analysis unit 121, a semantic analysis unit 122, an unnecessary expression extraction rule application unit 123, a content word extraction unit 124, a detection key production unit 125, an analysis dictionary 126 and an analysis grammar 127 utilized in the morphological, syntactic, and semantic analyses, an unnecessary expression extraction rule dictionary utilized by the unnecessary expression extraction rule application unit 123, and a related word dictionary 129 utilized by the detection key production unit 125.

Detailed Description Text (23):
Next, at the step 604, the unnecessary expression extraction rule specified by the unnecessary expression extraction rule dictionary 128 is applied at the unnecessary expression extraction rule application unit 123, to delete a partial structure which coincides with each unnecessary expression specified by the unnecessary expression extraction rule, and then at the step 605, the structure obtained up to the step 604 is stored into the individual data storage unit 16.

Detailed Description Text (24):
Next, at the step 606, the content words in the structure obtained up to the step 604 are extracted at the content word extraction unit 124, and then at the step 607, the detection key is produced at the detection key production unit 125 by using the content words extracted at the step 606 along with appropriate logic operators.

Detailed Description Text (25):
Next, at the step 608, additional detection keys are also produced at the detection key production unit 125 by looking up related words of each extracted content word and replacing each extracted content word in the detection key produced at the step 607 by each looked up related word. Then, all the content words in the detection key as well as their related words are set as detection target keywords in the detection keys.

Detailed Description Text (27):
As a concrete example of the result of the operation by this input analysis unit 12, FIG. 7 shows various results obtained at various stages in the flow chart of FIG. 6 described above, for a particular input sentence in Japanese (with English translation provided in parentheses). More specifically, for the input sentence indicated in (a) of FIG. 7, the morphological analysis result appears as indicated in (b) of FIG. 7 in which the input sentence is divided into words. Then, the syntactic analysis result appears as indicated in (c) of FIG. 7 in which the tree structure representing the syntactic structure of the input sentence is generated. Then, the semantic analysis result appears as indicated in (d) of FIG. 7 in which the tree structure representing the semantic structure of the input sentence is generated.

Detailed Description Text (28):
In the semantic analysis, the semantic analysis rule specified in the analysis dictionary 126 such as that indicated in FIG. 8 for example is applied. Namely, this semantic analysis rule indicated in FIG. 8 specifies the semantic structure corresponding to the syntactic structure in which the Japanese verb "mochiiru" (meaning "use") appears between a noun and a so called sahen-noun (i.e., a noun that can be turned into a verb by affixing a Japanese verb "suru" (meaning "do") at the end) as a so called instrument case relationship between the noun and the sahen-noun. The similar semantic analysis rules are also specified in the analysis dictionary 126 for the other frequently used verbs such as "employ", "invoke", etc. It is noted here that in FIGS. 7 and 8, "wo" indicates a Japanese postpositional word, "rentai" indicates a symbol for a participial adjective, "nitsuite" indicates a Japanese expression meaning "about", and "obj" indicates a symbol for an objective case relationship.

Detailed Description Text (29):
Then, the unnecessary expression extraction result appears as indicated in (e) of FIG. 7 in which a partial structure which coincides with the unnecessary expression specified by the unnecessary expression extraction rule in the unnecessary expression extraction rule dictionary 128 is deleted from the tree structure obtained by the semantic analysis unit 122. Here, as shown in FIG. 9 for example, the unnecessary expression extraction rule indicates the partial structure to be deleted as "-obj.fwdarw.KNOW". It is to be noted, however, that the unnecessary expression extraction rule may indicate a word to be deleted instead of the partial structure to be deleted.

Detailed Description Text (30):

Then, the content word extraction result appears as indicated in (f) of FIG. 7 in which the content words "example" and "machine translation" are extracted from the tree structure obtained by the unnecessary expression extraction rule application unit 123.

Detailed Description Text (31):
Finally, the detection key production result appears as indicated in (g) of FIG. 7 in which the detection key is formed by connecting the content words extracted by the content word extraction unit 124 with an appropriate logic operator "+".

Detailed Description Text (32):
Next, the detection processing unit 13 has a detailed functional configuration as shown in FIG. 10, which comprises: a keyword index matching unit 131 and a semantic structure matching unit 132 which are connected with the document storage unit 15 and the individual data storage unit 16, and a document file set calculation unit 133 connected with the keyword index matching unit 131 and the semantic structure matching unit 132.

Detailed Description Text (34):
First, at the step 1101, the detection key stored in the individual data storage unit 16 is taken out to the keyword index matching unit 131, and then at the step 1102, the detection using the taken out detection key is carried out on a keyword index memory in the document storage unit which stores all words of all the documents in the document database, so as to obtain those documents which contain the same words as those used in the detection key.

Detailed Description Text (36):
Then, at the step 1104, the semantic structure matching unit 132 Judges whether more than one documents are stored in the individual data storage unit 16 as the detected documents obtained at the step 1103 and the semantic structure of the input character string is stored in the individual data storage unit 16, or not. In a case of the affirmative, the semantic structure of the input character string is taken out from the individual data storage unit 16 at the step 1105, and the detection using the taken out semantic structure is carried out on a semantic structure index memory in the document storage unit 15 which stores all the semantic structures of all the documents in the document database, so as to obtain those documents which have the same semantic structure as that of the input character string, and then the result obtained at the step 1106 is stored in the individual data storage unit 16 at the step 1107. Otherwise, the steps 1105 to 1107 are skipped.

Detailed Description Text (39):
For example, for the Japanese keyword "kikai" (meaning "machine") formed by two kanji characters as indicated on the first line in FIG. 13, the first character registered at the address 00053 in the head character storage region has a link data "00935" specifying the second character of the keyword as that registered at the address 00935 in the subsequent character storage region. In addition, this second character at the address 00935 also has the link data "01201" specifying the third character, and the third character at the address 01201 has the link data "01309" specifying the fourth character, for another keyword "kikai-honyaku" (meaning "machine translation") formed by four kanji characters as indicated on the last line in FIG. 13 which contains the above described keyword "kikai" as a first part. Furthermore, the second character at the address 00935 also has a file data "file 4 (34, 35, 72, 86)" indicating that the keyword "kikai" is contained in the document data having the file name "file 4" at sentence No. 34, 35, 72, and 86. Similarly, the fourth character at the address 01309 has two file data "file 25 (18, 42)" and "file 21 (23)", indicating that the keyword "kikai-honyaku" is contained in the document data "file 25" at sentence No. 18 and 42 and the document data "file 21" at sentence No. 23. On the other hand, the first character at the address 00091 in the head character storage region is common to two keywords "jikken" (meaning "experiment") and "Jitsurei" (meaning "example") as indicated on the second and third lines in FIG. 13, so that it has two link data "01003" and "01004" specifying the respective second characters for these two keywords. In FIG. 12, an isolated "0" functions as a separator for separating the character, link data, and the file data. Also, the first characters of the keywords are registered in the continuous head character storage region in a sorted order such as that of JIS (Japanese Industrial Standard) codes.

Detailed Description Text (40):
Thus, the keyword index matching unit 131 looks up this keyword index memory to make a matching of each detection target keyword in the detection key, and obtains the documents containing the same detection target keyword as the detection key according to the document data registered for the matched keyword in the keyword index memory.

Detailed Description Text (41):
More specifically, the keyword index matching unit 131 and the document file set calculation unit 133 operate according to the flow chart of FIGS. 14A and 14B as follows.

Detailed Description Text (43):
Then, at the step 1302, whether "i" is greater than "N" or not is judged. Unless "i" is greater than "N" at the step 1302, next at the step 1303, the first character of the keyword "i" is detected in the head character storage region of the keyword index memory, and a block registering that first character is designated as a block "A". Here, as the head character storage region stores the first characters in a sorted order, so that the block registering the first character of the keyword "i" can be obtained easily by carrying out the binary search.

Detailed Description Text (50):
More specifically, the semantic structure index matching unit 132 operates according to the flow chart of FIG. 16 as follows.

Detailed Description Text (51):
First, at the step 1501, the semantic structure index memory is looked up for the target word in the semantic structure of the input character string. Then, only when the coinciding target word is found out in the semantic structure index memory at the step 1502, the relation symbol registered for the coinciding target word is looked up and matched with the relation symbol in the semantic structure of the input character string at the step 1503. Then, only when the coinciding relation symbol is found out in the semantic structure index memory at the step 1504, the source word registered for the coinciding relation symbol is looked up and matched with the source word in the semantic structure of the input character string at the step 1505. Then, only when the coinciding source word is found out in the semantic structure index memory at the step 1506, the document file names and sentence No. registered for the coinciding source word is stored into the individual data storage unit 16 along with the matched semantic structure itself.

Detailed Description Text (53):
The document structure analysis unit 141 analyzes the chapter structure of each document by extracting title data indicating the chapters and sections in each document. Here, the details of the document structure analysis to be carried out by this document structure analysis unit 141 is not essential to the present invention, and any known schemes can be adopted. For example, the scheme disclosed in Doi, M., et al.: "Research on Model Based Document Processing System DARWIN", Human-Computer Interaction-INTERACT '87, H. J. Bullinger, B, Shackel (Ed.), Elsevier Science Publishers B.V. (North-Holland), 1987, pp. 1101-1106, can be utilized here.

Detailed Description Text (54):
The text structure analysis unit 142 analyzes the logical structure of the sentences of each chapter or section by extracting rhetorical expressions used in the sentences of each chapter or section. Here, again, the details of the text structure analysis to be carried out by this text structure analysis unit 142 is not essential to the present invention, and any known schemes can be adopted. For example, the scheme disclosed in Sumita, K., et al.: "A Discourse Structure Analyzer for Japanese Text", Proceedings of the International Conference on Fifth Generation Computer Systems 1992, Institute for New Generation Computer Technilogy (Ed.), pp. 1133-1140, can be utilized here.

Detailed Description Text (56):
The text re-construction unit 144 generates a summary of each document from the key sentences obtained by the key sentence judgement unit 143.

Detailed Description Text (87):
The detection control unit 18 also determines the display priority order among the detected documents by using the detection result obtained from the detection key and the syntactic and semantic analyses results for the input sentences which are obtained by the detection processing unit 13 and stored in the individual data storage unit 16, and the summary data obtained by the summary generation unit 14 and stored in the individual data storage unit 16 as follows. Namely, the display priority order is determined according to the prescribed priority order conditions provided as a display priority rule dictionary in the detection control unit 18, as shown in FIG. 24. The detection control unit 18 then controls the detection result output unit 17 to display the titles of the detected documents in the determined display priority order as the

detection result.

Detailed Description Text (109):
As another example, FIG. 35 shows an exemplary case of the word "something" in the input sentence is converted into a symbol "?", which will be regarded as matching with arbitrary word in the detection processing. In FIG. 35, a symbol "goal" indicates the objective case relationship.

Detailed Description Text (111):
For example, FIG. 36 shows an exemplary case of the input sentence specifying the bibliographical matters of the desired document as those written by "M. Tanaka" and issued "since 1980". In response, the detection processing unit 13 carries out the detection processing for only those documents which have "M. Tanaka" as the author and the issue year not less than "1980", according to the bibliographical matter analysis rules shown in FIG. 37 which are provided in the detection processing unit 13. In this case, the detection processing is carried out by looking up the keyword index memory for the keyword "machine translation" obtained as the analysis result as indicated in FIG. 36. Here, the word "papers" is not used as a keyword because of the application of the unnecessary expression extraction rule as shown in FIG. 38. The procedure for carrying out the detection processing is substantially similar to that in the first embodiment described above.

Detailed Description Text (112):
Next, the fourth variation concerning the display priority order setting will be described. Namely, in the first embodiment described above, the display priority order is determined by weighting the detected documents in accordance with their summaries. In contrast, in this fourth variation, the detected documents are weighted according to the display priority scores determined in accordance with the document structure analysis result such as the title, table of content, index, and references of each document.

Detailed Description Text (128):
In this case, the input analysis unit 12 possesses a keyword extraction rule dictionary for specifying rules for extracting the content words, which has an exemplary form as shown in FIG. 43. Here, the user can freely modify, delete, and add the rules to this keyword extraction rule dictionary.

Detailed Description Text (131):
Then, the input analysis unit 12 extracts the content words from the input character string as the detection target keywords according to the morphological analysis result by using the keyword extraction rule dictionary and the unnecessary word dictionary. Here, according to the keyword extraction rule dictionary shown in FIG. 43, the word whose part of speech is noun or verb is extracted as a content word at the step 4402, and the unnecessary word dictionary is looked up for each extracted content word, and those content words not coinciding with the words registered in the unnecessary word dictionary are set as the detection target keywords at the step 4403.

Detailed Description Text (135):
Only when there is at least one document is stored in the individual data storage unit 16 and there is a sentence containing more than one keywords in said at least one document at the step 4408, the input analysis unit 12 takes out the morphological analysis result stored in the individual data storage unit 16 and carries out the syntactic analysis and the semantic analysis at the steps 4409 and 4410, respectively. Then, at the step 4411, the structure coinciding with that registered in the unnecessary expression extraction rule is deleted.

Detailed Description Text (139):
On the other hand, the Input analysis unit 12 takes out the morphological analysis result for the input character string stored in the individual data storage unit 16, and carries out the syntactic and semantic analyses for the input character string. Then, in a case the syntactic and semantic analysis results are obtained, the obtained syntactic and semantic analysis results are matched with the syntactic and semantic analysis results for the documents stored in the individual data storage unit 16, and the matching result is stored in the individual data storage unit 16 as the detection result.

Detailed Description Text (140):
Here, the detection result contains those which are not completely matching, and those which are not completely matching are accompanied by information indicating this fact

in the individual data storage unit 16. This information can be used at a time of setting the display priority orders, such that the document without this information is displayed before the document with this information among the documents in the same display priority order.

Detailed Description Text (141):
For example, when the semantic structure obtained from the input sentence is as indicated in (a) of FIG. 45, and there is a document containing an expression "design automation system using computer" which has the semantic structure as indicated in (b) of FIG. 45, such a document will be included in the detection result even though this semantic structure of (b) of FIG. 45 does not match the semantic structure of (a) of FIG. 45 completely, because this semantic structure of (b) of FIG. 45 contains all of the keyword "computer", the relation "instrument", and the keyword "design"0 relevant to the semantic structure of (a) of FIG. 45.

Detailed Description Text (148):
In this case, the detection control unit 18 stores the keywords and/or the semantic structures contained in the summary generated by the summary generation unit 14, in relation to the document from which the summary has been generated, in the document storage unit 15. Then, in a case the keywords and/or the semantic structures stored in relation to the document exist, the detection processing unit 13 carries out the keyword detection operation and/or the semantic structure detection operation by using these stored keywords and/or semantic structures.

Detailed Description Text (149):
Here, instead of storing the keywords and/or semantic structures in relation to the document, the summary keyword index memory and/or the summary semantic structure index memory may be produced and utilized in the detection processing.

Detailed Description Text (156):
In this fourteenth variation, the display means 4 in the overall configuration of FIG. 1 uses a bit map display in which the display screen can be divided such that a list of identifiers of the detected documents can be displayed simultaneously along the text content of the original document and the summary for a selected one of the detected documents. The user enters the input for selecting the desired one of the detected documents as well as command input for controlling the manner of detection result display, through the input means 6. Here, the detection processing unit 13 stores either the detected documents themselves or their identifiers as the detection result obtained by the detection processing operation in the individual data storage unit 16.

Detailed Description Text (159):
Also, in this fourteenth variation, the summary generated by the summary generation unit 14 contains various information such as a correspondence relationship between the original document and the summary, the title, author, chapter and section headers, etc. of the document, as well as the abstract of each chapter of the document.

Detailed Description Text (164):
Here, the detection control unit 18 also determines the display priority order among the detected documents according to the prescribed priority order conditions provided as a display priority rule dictionary in the detection control unit 18, similar to the first embodiment described above. The detection control unit 18 then controls the detection result output unit 17 to display a list of the identifiers of the detected documents in the determined display priority order as the detection result.

Detailed Description Text (186):
Then, at the step 4001, whether the character in the summary sentence has been selected or not is checked in order to distinguish this operation from the other input processing. When the character on the summary document is selected, next at the step 4002, the character position of the selected character is obtained. In this example, the character position is specified by a number of characters from the first character in the summary display to the selected character.

Detailed Description Text (190):
As a concrete example, FIG. 65 shows the original document display for the original document corresponding to the summary shown in FIG. 62, when "3. System function" is selected.

Detailed Description Text (209):
Next, the input analysis unit 12' has a detailed functional configuration as shown in

FIG. 68, which comprises: a morphological analysis unit 41, a content word extraction unit 42, and an unnecessary word dictionary 43 utilized by the content word extraction unit 42.

Detailed Description Text (212):
Next, at the step 6902, the content word is extracted from the input sentence at the content word extraction unit 42 according to the morphological analysis result. Then, at the step 6903, whether the content word extracted at the step 6902 exists in the unnecessary word dictionary 43 or not is determined, and only when the extracted content word is not in the unnecessary word dictionary 43, the extracted content word is set as the detection target keyword at the step 6904, whereas otherwise the step 6904 is skipped.

Detailed Description Text (213):
Then, whether there is any other content word in the input sentence or not is determined at the step 6905, and only when there is another content word in the input sentence, the operation returns to the step 6902 above to repeat the steps 6902 to 6904 for the next content word in the input sentence, until all the content words in the input are extracted.

Detailed Description Text (215):
Then, the content word extraction result appears as indicated in (c) of FIG. 70 in which the content words "topics", "translation", and "examples" are extracted from the morphological analysis result.

Detailed Description Text (217):
Next, the detection processing unit 13' has a detailed functional configuration as shown in FIG. 71, which comprises: a keyword index matching unit 71 connected with the input analysis unit 12' and the individual data storage unit 16, and a document file set calculation unit 72 connected with the keyword index matching unit 71.

Detailed Description Text (218):
The keyword index matching unit 71 carries out the detection operation for each keyword entered from the input analysis unit 12' on the document data in the document storage unit 15 to obtain a set of documents containing the same keyword. The document file set calculation unit 72 then carries out a logical set calculation for the documents obtained by the keyword index matching unit 71 for all the keywords to obtain the final document file set for the detected documents. Here, the dictionary storage unit 15 possesses the keyword index memory similar to that shown in FIG. 12 for the first embodiment described above, and this operation of the detection processing unit 13' can be carried out by the same procedure as that shown in FIGS. 14A and 14B for the first embodiment described above.

Detailed Description Text (224):
Namely, the document data may contain the technical terms of a particular field or special words defined within the document itself. In this regard, since the summary is generated by the summary generation unit 14 by selecting key sentences alone from the document, there is a possibility for the selected key sentences to contain such technical terms or special words without their definitions, as their definitions are given in the other non-selected earlier part of the documents, such that the user reading the summary can be unable to understand the meaning of such technical terms or special words. In order to remedy this situation, the internal document relation data analysis unit 21 extracts the definitions of the technical terms or special words from the document and relate them to the technical terms or special words used in the summary.

Detailed Description Text (233):
(1) $(noun phrase) ($(noun phrase))

Detailed Description Text (234):
(2) $(noun phrase) is called $(noun phrase)

Detailed Description Text (236):
In the above examples, the pattern (1) indicates the appearance of the noun phrase in parentheses immediately after another noun phrase, which corresponds to a type of definition expression in which the abbreviated expression is juxtaposed with the full expression. For example, in the above example of FIG. 73, "an advanced control rod drive mechanism (ACRDM) . . . " corresponds to this pattern (1).

Detailed Description Text (238):
When that preceding sentence matches with any of the prescribed definition expression patterns, that preceding sentence can be judged as the defining sentence of the noun which is found to be contained in that preceding sentence, for which the internal document relation data can be produced as described above. Then, when the defining sentence for a certain word is found out, the other sentences containing that certain word are searched out, and the internal document relation data are produced for the other sentences containing that certain word as well.

Detailed Description Text (246):
In this operation, the reference expression can be detected at the step 7503 by matching each sentence S with the prescribed reference expression patterns provided in table format. For example, the reference expression patterns can be specified as follows.

Detailed Description Text (254):
The detection result summary display unit 231 displays a list of the titles of the detected documents and the summary of the detected document with the highest display priority order as the initial screen display of the detection result.

Detailed Description Text (280):
It is to be noted that, in the above operation, the detection result summary display unit 231 can be easily modified such that the titles of the detected documents alone is displayed initially, and the summary for the document with the selected title is displayed only in response to the viewpoint selection event entered by the user.

Detailed Description Text (284):
When the full text detection operation is carried out by the detection processing unit 13', it becomes possible to identify a matching portion of the original document which contains the detection target word. In addition, by utilizing the text structure analysis result obtained by the summary generation unit 14 for the purpose of the summary generation, it also becomes possible to obtain data concerning the relation of each matching portion of the original document with respect to the entire document in view of the text structure.

Detailed Description Text (285):
Consequently, in this first variation, it is possible to rearrange the detected documents obtained at the detection processing unit 13' into a prescribed order determined according to the relations of the various matching portions.

Detailed Description Text (286):
For example, consider a case in which the detection result obtained by the detection processing unit 13' contains a set of detected documents {document 1, document 2, document 3, document 4}, and the relations of the matching portions in the respective detected documents are {serial relation (right), example (right), serial relation (right), serial relation (left)}, where left or right specified in the parentheses accompanying each relation indicates a left or right node of sentence which to which the matched detection target word belongs to in the text structure of the original document. In such a case, these detected documents can be rearranged into an order of {document 1, document 3, document 4, document 2} by grouping the documents with the relation of the serial relation together.

Detailed Description Text (288):
In this configuration of FIG. 82, the full text detection unit 281 carries out the full text detection operation according to the known full text detection algorithm, and the document file set calculation unit 282 carries out the set calculation operation similar to that of the steps 1312 to 1315 in the flow chart of FIG. 14B described above for the first embodiment. Then, the rearrangement unit 283 carries out the rearrangement operation for the detected documents obtained by the document file set calculation unit 282, by obtaining the relations of the matching portions for the detection target words according to the text structure analysis result obtained by the summary generation unit 14 and subsequently stored in the individual data storage unit 16, and then rearranging the order of the detected documents according to the relation order table 284.

Detailed Description Text (290):
In this first variation, the text structure analysis result for each document is expressed in the individual data storage unit 16 by a data structure as shown in FIG. 84, in which a block corresponding to each node has four data entries of "text

structure Rel" indicating the relation name, "text structure L" indicating a pointer to a left side branch, "text structure R" indicating a pointer to right side branch, and "text structure Level" indicating a path length from the root node (a number of arcs from the root node), in the tree structure of the document. In other words, in comparison with the data structure of FIG. 19 described above for the first embodiment, this data structure of FIG. 84 includes the "text structure Level" indicating the path length from the root node as an additional information.

Detailed Description Text (293):
Then, the matching relation extraction processing to 8503 and 8504, to obtain a set (Relation "i", Arc "i") for a document name "i" and a sentence number "i" at the step 8503, and to obtain a set (Relation "j", Arc "j") for a document name "j" and a sentence number "j" at the step 8504. Here, the sentence number "i" or "j" indicates the sentence number of the matching sentence containing the detection target word in the document "i" or "j".

Detailed Description Text (298):
The matching relation extraction processing to be carried out at the steps 8503 and 8504 is a sub-routine for receiving the document name and the sentence number as the input and returning the relation and the arc as the output, which is executed according to the flow chart of FIG. 86 as follows.

Detailed Description Text (302):
When the "text structure Level" of the block at the address "B" is greater than the prescribed threshold T at the step 8603, the address "B" is set as the address "A" at the step 8604, and the operation returns to the step 8602 described above, so as to trace back the tree structure toward the root node step by step.

Detailed Description Text (395):
To this end, the document selection processing unit 232 operates according to the flow chart of FIG. 107, which differs from that shown in FIG. 101 for the sixth variation described above in that there are additional steps of the step 8081 for displaying the original document corresponding to the displayed summary, and the step 8082 for adding the emphasis such as shading or reversed coloring to the display of sentences in the original documents which are corresponding to the summary sentences of the displayed summary, between the steps 8071 and 8022.

Detailed Description Text (398):
As a concrete example, with respect to the display summary as shown in FIG. 110A, the user can make the pointing action by using the cursor indicated as a black triangle specifying the chapter 1 in FIG. 110A. Then, FIG. 110B indicates the display of the corresponding original document resulting from the pressing of the "original" icon button in the state of FIG. 110A, in which the emphasis in a form of the reversed color is added to the displayed portions of two sentences which are contained in the displayed summary of FIG. 110A as the summary sentences. Here, the correspondence between the summary sentences and the sentences in the original documents can be recognized from the information concerning which sentence in the original document is each summary sentence originated from, that can be obtained at a time of selecting the sentences with the penalty below the threshold P2 at the text re-construction unit 144.

Detailed Description Text (402):
More specifically, the document structure analysis unit 141 carries out the additional operation according to the flow chart of FIG. 111 to extract the paragraph data concerning the paragraphs in the original documents, as follows.

Detailed Description Text (420):
After that, the summary generation unit 14 continues its summary generation operation to generate the other summaries for the other detected documents, and these other summaries are transferred to the detection result output unit 17' when their transfer is requested from the detection result output unit 17'.

Detailed Description Text (424):
In this configuration of FIG. 117, the summary generation unit 14 generates the summaries for all the documents stored in the document storage unit 15 in advance, and stores the generated summaries in the summary storage unit 25.

Detailed Description Text (427):
As described in detail above, according to the second embodiment and its variations,

the detection result can be displayed in the desired viewpoint specified by the user, so that it becomes possible to provide a document detection system capable of automatically preparing and displaying a document summary for each document in a viewpoint which is efficiently comprehensible for the user, considering the limited visual data processing ability of the human user, such that the user can make a judgement concerning the appropriateness of the detection result quickly. In addition, the summaries generated at a time of the detection result display are stored in correspondence to the original documents, so that there is no need to generate the summaries again in the subsequent operations and consequently the subsequent processing time can be reduced considerably.

CLAIMS:

1. A document detection system, comprising:

input means for entering a detection command containing detection keywords specified by a user;

document storage means for storing a plurality of detection target documents;

summary generation means for generating a summary of each detection target document stored in the document storage means, and extracting keywords of each detection target document from the summary of each detection target document; and

detection processing means for identifying those detection target documents stored in the document storage means whose keywords extracted by the summary generation means match with the detection keywords in the detection command entered at the input means as detected documents and for retrieving the detected documents.

2. The system of claim 1, further comprising:

input analysis means for determining a semantic structure of the detection command containing natural language expressions entered by a user; and

wherein the summary generation means also extracts a semantic structure of each sentence in the summary of each detection target document, and the detection processing means identifies those detection target documents whose keywords match with the detection keywords and whose summary contain the sentence structure extracted by the summary generation means which matches with the sentence structure of the detection command determined by the input analysis means.

WEST

☑ Generate Collection Print

L24: Entry 4 of 8 File: USPT Dec 10, 2002

DOCUMENT-IDENTIFIER: US 6493663 B1
TITLE: Document summarizing apparatus, document summarizing method and recording medium carrying a document summarizing program

Abstract Text (1):
A document summarizing apparatus generates a comprehensive summary on a group of documents of relatively diverse contents. The structure of documents specified to be processed is analyzed in a phrase analyzing unit to generate analytic trees describing the dependencies between words. An analytic tree scoring unit adds scores to the analytic trees in accordance with their importance. An analytic tree score accumulating unit accumulates scored trees to unify the trees expressing the same concept to increases the scores added to the unified analytic trees. A sentence synthesizing unit then selects the trees with higher scores from within the set of analytic trees stored in the analytic tree score accumulating unit to synthesize a summary from the selected analytic trees. The present invention allows less limitation to be applied to the documents to be processed, as well as a comprehensive summary to be generated.

Brief Summary Text (10):
(2) Generation of Sentences Based on the Extracted Meanings

Brief Summary Text (11):
A method of sentence-synthesis based on the extracted meanings is described in the paper of McKeown and Radev, "Generating Summaries of Multiple News Articles" SIGIR-95 (1995); one example thereof is SUMMONS (SUMMarizing Online NewS articles). This technology uses slots in a given template to be fulfilled with information extracted from a plurality of documents. The information embedded in the template will be used as the conceptual structure for generating a summary of a pattern matched with the syntax.

Brief Summary Text (16):
The document summarizing apparatus disclosed in the Japanese Published Unexamined Patent Application No. Hei 10-134066 gathers similar paragraphs (of online news of other news companies) to a specified paragraph (of online news). The gathered paragraphs are then disassembled to sentences to regroup similar sentences Here the similar sentences may be defined to have the number of pattern-matched words greater than a threshold value. For example, "Typhoon #5, landing in Kyushu" or "a large typhoon #5 lands in Kyushu", etc.

Brief Summary Text (19):
However, the technologies of the Prior Art suffers from the problems as follows: (1) The enumeration of keywords cannot indicate the relational dependencies between words, since words are appeared independently. The reader has to guess the meaning behind them from the sequential order of keywords and from a variety of knowledge thereon. In order to guess what the collection of documents would say, the reader is required to have some knowledge on the field of the subject or the knowledge on the event described in the collected documents. (2) The generation of sentences from the extracted meanings is definitively limited to a narrow class of documents to be processed. This method has the definitive paragraphs subjected, such as articles on an affair of terrorism ("who did attack what, where, when and how, the victims and demolished buildings are . . . "). A meaning template for each kind of affairs should be predefined. This method may be used only for articles on the same affair. However, it may not be applicable to a collection of documents gathered as the result of search or of clustering. (3) The synthesis of following-up articles deals with the parent article and following articles of the same affair. Therefore this method is not applicable to a group of documents gathered as the result of search or of clustering. (4) The synthesis of a plurality of sentences is applicable only to the articles on the same affair. Therefore this method

is not applicable to a group of documents gathered as the result of search or of clustering.

Brief Summary Text (27):
In the document summarizing method as disclosed in the present invention, when a plurality of documents are specified to be processed, analysis graphs will be generated from the sentences contained in the specified documents and a summary will be synthesized based on the analysis graphs with higher importance.

Drawing Description Text (11):
FIG. 9 is a schematic diagram illustrating the sorted result of a set of analytic trees;

Detailed Description Text (10):
The sentence analyzing unit 12 upon reception of a document summarizing command analyzes the syntax of sentences of the set of documents held in the input documents holding unit 11. There are well known methods for analyzing the syntax in the Prior Art. In general, the result of syntax analysis may be expressed as a tree structure. In the preferred embodiment, a syntax analysis system describing the dependency between words is adopted. The resulting analytic tree 12a will be passed to the analytic tree expanding unit 13 and analytic tree scoring unit 14.

Detailed Description Text (11):
The analytic tree expanding unit 13 extends the analytic tree 12a generated by the phrase analyzing unit 12 to generate respective analytic tree 13a for each of sentences contained in the analytic tree 12a. The analytic tree generated by extending an analytic tree is referred to as a "subtree" hereinbelow. The set of analytic trees 13a generated as subtrees will be passed to the analytic tree scoring unit 14. The subtrees will be generated down to the minimal unit carrying a conceptual meaning. The minimal unit of analytic tree may be comprised of two nodes and an arc connecting these nodes.

Detailed Description Text (15):
In order to generate a summary in the document summarizing apparatus as described above, a group of documents 11a to be processed are stored in the input documents holding unit 11. Then the user of the document summarizing apparatus inputs a summary generating command to the document summarizing apparatus. The phrase analyzing unit 12 retrieves sequentially the documents stored in the input documents holding unit 11 one by one, and analyzes the syntax of the sentences contained in the retrieved documents to generate analytic tree 12a. Then, the analytic tree expanding unit 13 generates subtrees from the analytic tree 12a generated by the sentence analyzing unit 12. The analytic tree 12a generated by the sentence analyzing unit 12 and the analytic trees 13a generated as the subtrees by the analytic tree expanding unit 13 will be passed to the analytic tree scoring unit 14. These analytic trees will be scored in the analytic tree scoring unit 14. The analytic trees thus scored will be accumulated in the analytic tree score accumulating unit 15. When storing in the accumulator, if there already exists a same analytic tree, then these trees will be merged to accumulate their scores. This allows higher scores to be given to the common trees shared by a plurality of documents. Thereafter, the sentence synthesizing unit 16 selects the trees with higher scores to synthesize sentences therefrom. Thus synthesized sentences will be output as the summary 16a.

Detailed Description Text (16):
Now referring to FIG. 3, there is shown a flowchart illustrating the process of sentence synthesis according to the present invention; this process may be done in the sentence synthesizing unit 16. [S1] sorts in the order of higher score the analytic trees stored in the analytic tree score accumulating unit 15; [S2] extracts the topmost analytic tree; [S3] determines whether or not the extracted topmost tree is a subtree of analytic tree having a sentence assembled. If yes, then the process proceeds to step S2, otherwise to step S4; [S4] assembles a sentence from the extracted tree [S5] determines whether the process satisfies the termination condition. The termination condition is preselected as the number of analytic trees having sentence assembled, the number of characters of the assembled sentence, and so on. If yes then the process proceeds to step S6, otherwise to step S2; [S6] outputs a summary made from the generated sentences.

Detailed Description Text (29):
When the analytic trees and their subtrees have been generated from the documents stored in the input documents holding unit 11 and stored in the analytic tree score accumulating unit 15, the sentence synthesizing unit 16 will sort by the score these

trees accumulated in the analytic tree score accumulating unit 15.

Detailed Description Text (30):
Now referring to FIG. 9, there is shown a schematic diagram illustrating the sorted result of a set of analytic trees. The trees 43a to 43i of the set of trees 43 shown in FIG. 8 are listed in the order of their scores. The sentence synthesizing unit 16 will pick up the topmost relational dependency among the sorted trees to assemble a sentence from thus picked-up trees.

Detailed Description Text (31):
Now referring to FIG. 10, there is shown a schematic diagram illustrating the generation of a summary from the analytic tree. This is an example of generating a summary 51 from the first tree 43b shown in FIG. 9. In this example, the words are picked up by their order described in the record of the tree 43b, to convert the relational dependency into the adjunct (the relation corresponds to the adjunct in this case) to add to the word of the record. Then the summary 51 will be generated by connecting these phrases.

Detailed Description Text (33):
The sentence synthesizing unit 16 will select second topmost to iteratively generate another sentence. In the sentence synthesizing unit 16, if a selected tree is a subtree of an analytic tree already selected to generate a sentence, this tree will be omitted. Otherwise, if a subtree of a selected analytic tree is in the selected and generated tree, then this tree will not be omitted. In FIG. 9, the third tree 43c is a subtree of first analytic tree 43b, matching to the former criteria, therefore no sentence will be generated.

Detailed Description Text (34):
From second tree 42e, a sentence "Hoobar-wa Simon-City-History-Art-Museum-wo shien (support)" will be generated. The third analytic tree 43c will be omitted. From the fourth tree 43g, a sentence "Hoobar-wa satellite portable phone-wo kenkyuu(study)" will be generated. The fifth tree 43h will be omitted because this is a subtree of the fourth 43g. With respect to the sixth tree 43a, the first tree 43b is its subtree. This matches to the latter criteria of omission so that the sixth tree will not be omitted to generate a summary. Then a sentence will be generated as "Hoobar 50 g portable phone sell".

Detailed Description Text (40):
Now second preferred embodiment of the present invention will be described in detail below. In second preferred embodiment the importance score of words are used. Here each of words will not only be scored but also the importance level of the words contained in the analytic trees extracted as summary may be decreased to re-score the trees each time a summary is generated.

Detailed Description Text (46):
The sentence synthesizing unit 76 synthesizes sentences after the generation of analytic trees and the scoring thereof for all the input documents. In the sentence synthesizing process, the analytic trees stored in the analytic tree score accumulating unit 75 will be sorted by their score. Then the topmost tree among the group of sorted trees will be picked up to be eliminated from the analytic tree score accumulating unit 75. Then the words used will be stored in the word list 76a. The sentence synthesizing unit 76 will check the condition of termination. If the process does not terminate, then the unit will output the re-scoring request to the analytic tree scoring unit 74. When the re-scoring finishes, the analytic trees in the analytic tree score accumulating unit 75 will be sorted and the topmost will be again picked up.

Detailed Description Text (48):
When the analytic trees generated from all of the documents in the input documents holding unit 71 are stored in the analytic tree score accumulating unit 75, the sentence synthesizing unit 76 sorts the trees in the order of scores. Then the synthesizing unit 76 picks up the topmost tree to assemble a sentence, and stores the words contained in the tree in the word list 76a. The picked-up tree will be eliminated from the analytic tree score accumulating unit 75. At this point, if the terminating condition preset to the sentence synthesizing unit 76 in not satisfied, the sentence synthesizing unit 76 will send a re-scoring request to the analytic tree scoring unit 74. The information on the words immediately added to the word list 76a will be passed to the analytic tree scoring unit 74, which unit in turn, in response to reception of the re-scoring request, will decrease, for example divided by five, the importance score of words immediately added to the word list 76a. The updated importance score of

words will be used for calculating the score of analytic trees stored in the analytic tree score accumulating unit 75 to add the resulting score to the corresponding tree in the analytic tree score accumulating unit 75.

Detailed Description Text (76):
The relational table holding unit 110 holds as a relational table the rules of translation of the relational dependencies that may result in some similar meanings when replacing the adjunct ("HA-NOMINATIVE" or "GA-NOMINATIVE") between words in accordance with given rules.

Detailed Description Text (77):
The analytic tree score accumulating unit 105 translates the relational dependency in the stored analytic trees based on the relational table to determine which trees make a pair of same analytic trees. The unit decreases the score of one analytic tree according to the translation strategy before adding to the score of another analytic tree. More specifically, if there is a pair of trees that have the same nodes indicating words and different arcs indicating relational dependencies between words, the unit determines the similarity between the arcs by using the relational table. If it is determined to have similarity, the unit will translate the relational dependencies therebetween to merge to the analytic tree of similar meaning. The score will be added to the analytic tree of similar meaning by decreasing the original score of the tree to be merged.

Detailed Description Text (86):
The analytic tree score accumulating unit 105 selects the former tree 125 of these two relational dependencies for merging. In order not to extract more than one sentence of the same meaning, the analytic tree 126 that was not selected will be eliminated from the analytic tree score accumulating unit 105.

Detailed Description Text (91):
Now referring to FIG. 18, there is shown an example of relational table for verbs. In the relational table 112, translation rules are corresponding to subjective verb. This table is composed as follows:

Detailed Description Text (92):
First row: general translation rules. "NI-DATIVE" may be translated to "NITAISHITE-DATIVE" or "HE-DATIVE", with respective rate of 0.5 and 0.3.

Detailed Description Text (96):
When the analytic tree score accumulating unit 105 translates the dependencies by referring to the relational table 112, it selects the dependency before translation from the relational table 112 then to conform to the depended verb. If there is another verb that conforms to the depended verb, it will use the list of candidates in that line. Otherwise, it will use the line of generic rules (marked as "-").

Detailed Description Text (97):
It may be conceivable to alter the rules in accordance with the structure.

Detailed Description Text (98):
Now referring to FIG. 19, there is shown an example of a relational table based on the natural language structure. A relational table 113 conforms to the structure of natural language of the target, with rules and rates being defined. The table is composed as follows:

Detailed Description Text (105):
When the analytic tree score accumulating unit 105 translates the dependencies, it refers to the attribute appended to the analytic tree in order to invoke the rules of, for example, the first and second columns in the case of the passive voice. At this point, dependencies will be rearranged such that, if there are "GA-NOMINATIVE" and "NI-DATIVE" then the rule of the first row will be applied to translate them into "WO-ACCUSATIVE" and "GA-NOMINATIVE" respectively so as to convert an passive attribute to an active attribute. At the same time, their respective depending words will be rated. If there are "GA-NOMINATIVE" and "KARA-FROM" then the rule of the second row will be applied.

Detailed Description Text (106):
Now considering the case of applying the rule of the fifth row. It is assumed that the verb it labeled as intransitive or transitive in the result of analysis. The correspondence of intransitive and transitive is assumed to be described in the

dictionary used in the dependency analysis and to be labeled as an attribute of the verb in the analysis result. For example, when analyzing a sentence "cause a syntax error", the attribute Type: transitive Intransitive: occur is described to the verb "cause". If the rule in fifth row is applied to this sentence, the following operations should be performed: (1) change the dependency adjunct WO-ACCUSATIVE to GA-NOMINATIVE in "cause a syntax error WO-ACCUSATIVE"; (2) rate the score of "syntax error" by 0.5; and (3) replace "cause" with "occur".

Detailed Description Text (110):
In order to generate a summary of documents written in English according to the preferred embodiment, documents are to be analyzed similarly to the preceding embodiments. In this preferred embodiment, documents are analyzed based on the Lexical Functional Grammar (LFG) proposed by Bresnan in 1982. Now referring to FIG. 20, there is shown an example of document analyzed according to LFG. The analysis result from the syntax analysis of the documents according to LFG is referred to as feature structure, expressed as a list of pairs of attributes and its values. In FIG. 20, one single elementary structure is formed between "[" and "]". In the left columns of the elementary structure attributes of each element forming the document are placed, and in the right columns spaced apart from the left columns, values corresponding to the attributes listed in its left; the values corresponding to the attribute may be a string of characters, or another elementary structure comprised of attributes and values. An upper arrow ".uparw." within parentheses positioned immediately after the word in a value indicates that the word takes as an argument the value corresponding to the element located in the same parenthesis as this upper arrow. If the value corresponding to each attribute is shared, a common code such as [1] is placed at all locations of the shared value, except for only one location into which the shared value is described. In FIG. 20, some attributes such as tense (TENSE) and number (NUM) are omitted for the sake of simplicity.

Detailed Description Text (121):
After accumulating scores of subgraphs, a subgraph having the highest score will be picked up, in a similar manner to the preceding first through fourth preferred embodiments above, synthesized to generate a summary. It should be noted that in this embodiment another subgraph synthesizing method will be required which should be different from the preceding first to fourth embodiments generating a summary from documents written in Japanese, because a summary for documents written in English should be generated.

Detailed Description Text (123):
For synthesizing subgraphs that have been retrieved, a type of summary to be generated may be specified. The types of summary include a verbal phrase type (V), a noun phrase type (NP), a gerund type (V-ing) and so on. The user of the document summarizing apparatus in accordance with the preferred embodiment may arbitrarily select one appropriate style.

Detailed Description Text (124):
After the selection of summary style, the pattern for generating summary may be specified. The sentence synthesizing unit in the document summarizing apparatus in accordance with the preferred embodiment contains a pattern dictionary that stores a variety of elementary structures, styles, and generating patterns, and is indexed by the PRED element, so as to search in the pattern dictionary with the elementary structure and selected style of the retrieved subgraph to determine a pattern to be generated. For the pattern dictionary search, the PRED element of the retrieved subgraph should be obtained to refer to the index of pattern dictionary with the retrieved PRED element to obtain a set of entries of pattern dictionary corresponding to the index matched to the retrieved PRED element.

Detailed Description Text (125):
After the extraction of a set of entries of pattern dictionary, the elementary structure of the entries extracted from the pattern dictionary will be unified with the elementary structure of the extracted subgraphs to extract therefrom the entries of pattern dictionary having the elementary structure matched to the retrieved subgraphs. Then the style of the entries of pattern dictionary having a matched elementary structure will be compared with the selected style to further extract therefrom the entries of pattern dictionary having the same style and the same elementary structure as the extracted subgraphs.

Detailed Description Text (126):
By picking up the pattern to be generated of the entries of pattern dictionary having

thus extracted, value of elements such as arg1, arg2, . . . etc. of the extracted subgraphs will be substituted into the "element", which is the part having an argument in the pattern to be generated. If the pattern substituted with the value of elements arg1, arg2, . . . contains a character string, the string may be held as a partial string. Otherwise if the pattern substituted with the value of elements contains another elementary structure, then the matching to the pattern dictionary as described above will be recursively applied to ultimately obtain some partial strings. Thereafter, a summary may be generated by concatenating partial strings thus obtained.

Detailed Description Text (131):
Now assuming that the elementary structure 162a and 162b match to the elementary structure of normalized pattern 150 as the result of unification of the elementary structures in the pattern dictionary entry 161 with the elementary structure of the normalized pattern 150, i.e., with the retrieved subgraph. As shown in FIG. 24A, there are two styles, "S" and "NP", which correspond to the elementary structure 162a, 162b in the pattern dictionary entry 161. In this specific example "NP" is selected for the styles so the pattern of the pattern dictionary entry matching with the elementary structure and style of the normalized pattern 150 of the retrieved subgraph will be the pattern 164c, indicated as `(POSSESS arg1) construction of (NP arg2)`.

Detailed Description Text (134):
Now referring to the flowcharts shown in FIG. 25 and FIG. 26, the method for synthesizing subgraphs will be further described. [S10] put into F an input elementary structure of the subgraph having the highest score, and put into S the selected style; [S11] retrieve PRED element from F; [S12] select matching PRED elements from the pattern dictionary to put into S1; [S13] unify the elementary structures of S1 with F to put into S2 if successful; [S14] retrieve from within S2 the pattern P of which the style matches to S; [S15] substitute null string into the string "s" that is the final summary to be generated in order to initialize the string "s" [S16] determine whether or not P contains "elements" with an argument. If P contains no "element" then the step proceeds to S17, otherwise, if P contains "elements" then the step proceeds to S18; [S17] output the string "s", which has strings of P concatenated: [S18] retrieve one element "p" from P to substitute its argument; [S19] determine whether or not substituted "p" contains still another elementary structure. If "p" contains another elementary structure, then the step proceeds to S20, otherwise, if it contains no elementary structure, then the step proceeds to S24; [S20] retrieve another element "p" having another elementary structure; [S21] retrieve the specified style; [S22] recursively apply the algorithm from S10 to S24; [S23] concatenate the result obtained in step S22 to "s"; [S24] concatenate "p" to "s".

Detailed Description Text (135):
As can be appreciated from the description above, a summary may be generated by analyzing the structure of documents written in English, generating analysis graphs from the analysis results, extracting subgraphs therefrom, adding scores to the extracted subgraphs, picking up a subgraph having the largest accumulated score, and checking thus retrieved subgraph with a Pattern dictionary, allowing to yield the effect for the documents written in English identical to the first to fourth embodiments above for Japanese documents.

Detailed Description Text (137):
The pattern dictionary may also be structured by eliminating "pattern to be generated" and putting superficial cases into the elementary structures. In such a structure, if unified successfully, a superficial phrase will be provided in the elementary structure.

CLAIMS:

5. The document summarizing apparatus according to claim 1, further comprising: a relational table holding unit that holds a relational table defining translation rules for translating relational dependencies between words without altering the meaning of sentence, wherein said analysis graph score accumulating unit detects a pair of analysis graphs that ultimately results in the identical analysis graph when said analysis graphs are translated in compliance with the relational table held by said relational table holding unit so as to subtract the score of one analysis graph of the pair in response to the translation level and to add that score to the score of another analysis graph of the pair.

WEST

☑ Generate Collection Print

L24: Entry 3 of 8 File: PGPB May 30, 2002

DOCUMENT-IDENTIFIER: US 20020065857 A1
TITLE: System and method for analysis and clustering of documents for search engine

Abstract Paragraph (1):

A system and method for searching documents in a data source and more particularly, to a system and method for analyzing and clustering of documents for a search engine. The system and method includes analyzing and processing documents to secure the infrastructure and standards for optimal document processing. By incorporating Computational Intelligence (CI) and statistical methods, the document information is analyzed and clustered using novel techniques for knowledge extraction. A comprehensive dictionary is built based on the keywords identified by the these techniques from the entire text of the document. The text is parsed for keywords or the number of its occurrences and the context in which the word appears in the documents. The whole document is identified by the knowledge that is represented in its contents. Based on such knowledge extracted from all the documents, the documents are clustered into meaningful groups in a catalog tree. The results of document analysis and clustering information are stored in a database.

Summary of Invention Paragraph (15):

[0013] 3. Multi-search tools: These tools usually pass the request to several search engines and prepare the answer and one (combined) list. These services usually do not have any "indexes" or "spiders"; they just sort the retrieved information and eliminate redundancies.

Summary of Invention Paragraph (20):

[0018] Phrase search--retrieval of documents which include a sequence of words or a full sentence provided by a user usually between delimiters;

Summary of Invention Paragraph (26):

[0024] During the presentation of the results, apart form the list of hits (Internet links) sorted in appropriate ways, the user is often informed about the values of additional parameters of the search process. These parameters are known as precision, recall and relevancy. The precision parameter defines how returned documents fit the query. For example, if the search returns 100 documents, but only 15 contain specified keywords, the value of this parameter is 15%. The recall parameter defines how many relevant documents were retrieved during the search. For example, if there are 100 relevant documents (i.e., documents containing specified keywords) but the search engine finds 70 of these, the value of this parameter would be 70%. Lastly, the relevance parameter defines how the document satisfies the expectations of the user. This parameter can be defined only in a subjective way (by the user, search redactor, or by a specialized IQ program).

Summary of Invention Paragraph (29):

[0027] With the recent Internet boom, the number of servers has risen to more than 18 million. The number of domains has grown from 4.8 million in 1995 to 72.4 million in 2000. The number of web pages indexed by search engines has risen from 50 million in 1995 to approximately 2.1 billion in 2000. Meanwhile, the deep Web, with innumerable web pages not indexable by search engines, has grown to about 17,500 terabytes of information consisting of over 500 billion documents. Obviously, advanced mechanisms are necessary to discover all this information and extract meaningful knowledge for various target groups. Unfortunately, the current search engines have not been able to meet these demands due to drawbacks such as, for example, (i) the inability to access the deep Web, (ii) irrelevant and incomplete search results, (iii) information overload experienced by users due to the inability of being able to narrow searches logically and quickly, (iv) display of search results as lengthy lists of documents that are laborious to review, (v) the query process not being adaptive to past query/user sessions, as well as a host of other shortcomings.

Summary of Invention Paragraph (32):
[0029] According to the invention, a method for analyzing and processing documents is provided. The method includes the steps of building a dictionary based on keywords from an entire text of the documents and analyzing text of the documents for the keywords or a number of occurrences of the keywords and a context in which the keywords appear in the text. The method further includes clustering documents into groups of clusters based on information obtained in the analyzing step, wherein each cluster of the groups of clusters includes a set of documents containing a same word or phrase.

Summary of Invention Paragraph (33):
[0030] In embodiments, the groups of clusters are split into subclusters by finding words which are representative for each of the group of clusters and generating a matrix containing information about occurrences of the top words in the documents from the groups of clusters. New clusters are then created based on the generating step which corresponds to the top words and a set of phrases. The splitting may be based on statistics to identify best parent cluster and most discriminating significant word in the cluster. In further embodiments, the clustering may be performed recursively and may additionally include creating reverted index of occurrences of words and phrases in the documents, building a directed acyclic graph and counting the documents in each group of clusters. The clustering may further include generating document summaries and statistical data for the groups of clusters, updating global data by using the document summaries and generating cluster descriptions of the groups of clusters by finding representative documents in the each cluster of the groups of clusters. The clustering may also include finding elementary clusters associated with the groups of clusters which contain more than a predetermined size of the documents.

Summary of Invention Paragraph (34):
[0031] The analyzing step may also include analyzing the documents for statistical information including word occurrences, identification of relationships between words, elimination of insignificant words and extraction of word semantics, and is performed on only selected documents which are marked. The analyzing step may also include applying linguistic analysis to the documents, performed on titles, headlines and body of the text, and content including at least one of phrases and the words. The analyzing step may also include computing a basic weight of a sentence and normalizing the weight with respect to a length of the sentence. Thereafter, ordering the sentences with the highest weights in an order which they occur in the input text and providing a priority to the words by evaluating a measure of particular occurrence of the words in the documents. The keywords may then be extracted from the documents which are representative for a given document.

Detail Description Paragraph (13):
[0073] The presentation layer 1020A generates WebPages and includes dynamic content in the webpage. The dynamic content typically originates from a database (e.g. a list of matching products, a list of transaction conducted over the last month, etc.) Another function of the presentation layer 1020A is to "decode" the WebPages coming back from the client (e.g. find the user-entered data and pass that information onto the business logic layer). The presentation layer 1020A is preferably built using the Java solution using some combination of Servlets and JavaServer Pages (JSP). The presentation layer 1020A is generally implemented inside a Web Server (like Microsoft IIS, Apache WebServer, IBM Websphere, etc.) The Web Server can generally handle requests for several applications as well as requests for the site's static WebPages. Based on its initial configuration, the web server knows which application to forward the client-based request (or which static webpage to serve up).

Detail Description Paragraph (23):
[0083] The data layer 1030 is responsible for managing the data. In a simple example, the data layer 1030 may simply be a modem relational database. However, the data layer 1030 may include data access procedures to other data sources like hierarchical databases, legacy flat files, etc. The job of the data layer is to provide the business logic layer with required data when needed and to store data when requested. Generally speaking, the architect of FIG. 3 should aim to have little or no validation/business logic in the data layer 1030 since that logic belongs in the business logic layer. However, eradicating all business logic from the data tier is not always the best approach. For example, not null constraints and foreign key constraints can be considered "business rules" which should only be known to the business logic layer.

Detail Description Paragraph (31):

[0091] The Data Preparation module 200 analyzes and processes documents retrieved by the Data Acquisition module 100. The function of this module 200 is to secure the infrastructure and standards for optimal document processing. By incorporating Computational Intelligence (CI) and statistical methods, the document information is analyzed and clustered using novel techniques for knowledge extraction (as discussed below).

Detail Description Paragraph (32):
[0092] A comprehensive dictionary is built based on the keywords identified by the algorithms from the entire text of the document, and not on the keywords specified by the document creator. This eliminates the scope of scamming where the creator may have wrongly meta-tagged keywords to attain a priority ranking. The text is parsed not merely for keywords or the number of its occurrences, but the context in which the word appeared. The whole document is identified by the knowledge that is represented in its contents. Based on such knowledge extracted from all the documents, the documents are clustered into meaningful groups (as a collective representation of the desired information) in a catalog tree in the Data Preparation Module 200. This is a static type of clustering; that is, the clustering of the documents do not change in response to a user query (as compared to the clustering which may be performed in the Dialog Control module 300). The results of document analysis and clustering information are stored in a database that is then used by the Dialog Control module 300.

Detail Description Paragraph (35):
[0095] In particular, FIG. 6 describes the sequence of steps for the analysis-clustering process. In step 605, the process creates a thematic catalog of documents on the basis of a pre-selected thematic structure of Web pages. In step 610, the documents from the selected structure, and the words contained therein, are analyzed for statistical information such as, for example, documents and word occurrences, identification of relationships between words, elimination of insignificant words, and extraction of word semantics. The step 610 may also construct an inter-connection (link) graph for the documents. In step 615, the analyzed Web catalog documents are then grouped into larger blocks, e.g., clusters. The clusters are constructed into a hierarchical structure based on pre-calculated data (discussed in greater detail below). In step 620, the documents are then analyzed. Similar to the analysis and clustering processes for the structure of documents, the source documents taken from the Internet and other sources are also analyzed and clustered in a recursive manner, in step 625, until there is no new document detected at the source. This sequence of steps for the analysis-clustering process (FIG. 6) is an option, and there is no need to use pre-selected thematic structure of Web pages.

Detail Description Paragraph (37):
[0097] In more specificfity, the Data Preparation module is, in embodiments, divided into two separate analytical modules, DC1 and DC2 modules. The DC1 module processes the HTML documents downloaded by the spider, tags the documents and computes statistics used thereafter. Two main stages of analysis are called analyzer and indexer, respectively. The dc1 analysis is implemented, in embodiments, using Java and Oracle 8 database with the Oracle InterMedia Text option. InterMedia may help clustering (with its reverted index of word and phrase occurrences in documents).

Detail Description Paragraph (38):
[0098] The DC2 module processes the HTML documents downloaded by the spider and generates for the documents specific tags such as, for example, the document title, the document language and summary, keywords for the document and the like. The procedure of automatic summary generation comprises assigning weights to words and computing the appropriate weights of sentences. The sentences are chosen along the criterion of coherence with the document profile. The purpose of both modules is to group documents by means of the best-suited phrase or word when it is not possible to find association-based clusterings in the clusters obtained on the stage of dc1 analysis.

Detail Description Paragraph (47):
[0107] The HTML documents are also stored temporarily in a separate statistics database 803. The data gathered in this database is processed further by the indexer process which applies linguistic analysis of documents form (titles, headlines, body of the text) and its content (phrases and words). The indexer is also capable of upgrading a built-in dictionary which generates words that describe the document contents, creates indexes for documents in the database, associates the given document with other documents to create the concept hierarchy, clusters the documents using a tree-structure of concept hierarchy and generates a best-suited phrase for cluster description plus five most representative documents for the cluster. For the purpose of

further processing (taxonomy builders) the following statistics my be generated:

Detail Description Paragraph (48):
[0108] 50 best words or phrases for each document,

Detail Description Paragraph (56):
[0116] SUMMARY--the document summary

Detail Description Paragraph (60):
[0120] A Taxonomy (Skowron-Bazan) module 801 and a Taxonomy (Skowron 2) module 802 group documents by means of the best-suited phrase or word when it is not possible to find association-based clusterings in the clusters obtained on the stage of DC1 module 200A. The Skowron-Bazan Taxonomy Builder 800 is based on the idea of generation word conjunction templates best-suited for grouping documents. The Skowron 2 Taxonomy Builder 802, on the other hand, is based on the idea of approximative upper rough-set coverage of concepts of the parent cluster in terms of concepts appearing in the child cluster. The Skowron-Bazan Taxonomy Builder 800 is thus suited for single-parent hierarchies and the Skowron-2-Taxonomy Builder 802 allows for multiple-parent hierarchies.

Detail Description Paragraph (61):
[0121] The Skowron-Bazan Taxonomy Builder 800 comprises two processes: matrix and cluster. The matrix process generates a list of best words (or phrases) for each cluster and their occurrence matrix for the documents in the given cluster. Then the templates related to a joint appearance of words (or phrases) are computed by the cluster process and the tree-structure of taxonomy is derived from them. The cluster process splits too big word-asscociation -based clusters into subclusters using these statistics to identify best parent cluster and most discriminating significant words. The Skowron-2 Taxonomy Builder 802, on the other hand, comprises a completer and taxonomer process. The completer process adds new clusters based on word occurrence statistics, improving document coverage with clusters beyond word-association clustering. The taxonomer process splits clusters that were marked by the completer as too large. The taxonomer derives the subcluster from best characterizing words of the cluster and all its parents. (The functions associated with the matrix, taxonomer and completer processes are discussed in more detail below.)

Detail Description Paragraph (62):
[0122] The Dialogue module 300 assists the user in an interactive process of scanning the resources for the desired information. Also, some additional functions are supplied as preference configuration (Settings) and session maintenance. The Dialogue module 300 processes the query entered by the user and retrieves from the Database the appropriate tree-hierarchy. The hierarchy is the answer for the user-query and the dialog module makes its searching comfortable and efficient. The Dialogue module 300 also supports visualization of tags generated by the DC1 and DC2 modules 200A and 200B, respectively. Given a word, a phrase or an operator query, the Dialog module 300 groups the found documents into clusters labeled by the appropriate phrases narrowing the meaning of query.

Detail Description Paragraph (64):
[0124] Token: words/phrases found in documents during indexing them. Tokens may be single words, phrases or phrases found using simple heuristics (e.g., two or more consecutive words beginning with capital letters). Tokens may be used interchangeably with "word or phrase".

Detail Description Paragraph (68):
[0128] Indexing: a process of extracting all tokens found in a set of documents, and finding for each token documents containing the token. This information may be stored in a data structure called index.

Detail Description Paragraph (69):
[0129] Gist: a summary for documents both from the general point of view or form the point of view of a given theme.

Detail Description Paragraph (74):
[0133] FIG. 9 shows a case diagram implementing steps of the overall design of the search engine. Specifically, the admisintrator beings processinging at block 900. The processing includes setting processes running at block 900A and stopping at block 900B as well as setting the process paratmeters at block 900C and monitoring the processes at block 900D. The monitoring of the processes may be saved in logs at block 900E. The

data is prepared at block 901 which includes processing the documents at block 902 as well as analyzing the documents at block 904 and clustering the documents at block 906. Analyzing the documents includes extracting document meta information at block 904A for the clustering and processing. Meta information from HTML documents may include title, links, description and keywords. The clustering the documents includes generating a cluster heirarchy at block 906A, generting cluster descriptions at block 906B and assigning documents to elementary clusters at block 906C. the clustering description includes words or phrases that generate the cluster, the number of the documents in the cluster and, in embodiments, five documents that best represents the cluster. The generation of cluster hierarchy is preferably in the form of a connected directed acyclic graph. The limitations and requirements include:

Detail Description Paragraph (77):
[0136] Each cluster should be defined as a set of documents containing the same word or phrase.

Detail Description Paragraph (79):
[0138] FIG. 11 shows a flow diagram for the example shown in FIG. 10. Specifically, in block 1102, the documents are analyzed (preprocessing). In block 1104, the documents are processed. Processing is responsible for the extraction of valuable information from the documents. The processing is different from analysis in two aspects: it assumes more complex analysis of the document content and it is (intentionally) independent of clustering. in block 1106, the documents are clustered. In block 1108, the data preparation is completed.

Detail Description Paragraph (82):
[0141] FIG. 14 is a flow diagram showing an analysis of the documents using the DC1 analysis. The DC1 analysis is used for fast documents preprocessing. This is performed for preparing documents to be looked for by the dialog. The document is preferably stored in two forms: (i) the pre-processed form with HTML tags removed (used by the dialog when searching information required by the user) and (ii) the original HTML form stored for indexing. It is noted that extracting the document meta information and plain text content activities may be realized as a single activity.

Detail Description Paragraph (83):
[0142] In step 1402 of Figure, the documents are obtained from the package. In this step, the memory cache is used to limit the number of database connections openings. In step 1404, the documents content and the plain text are extracted. In this step, key algorithm 1 is used which may run concurrently for many documents. In step 1406, the documents HTML meta information is extracted using key algorithm 2. The extraction may include the content of title, links, meta keywords and meta description tags, and may run concurrently for many documents. In step 1408, the plain text version of a document with meta information tags is stored using key algorithms 1 and 2. In step 1410, further original documents may be stored for further processing. In step 1412, a decision is made as to whether there are any further documents. If not, then the process ends at step 1414. If there are further documents, then the process returns to step 1402.

Detail Description Paragraph (84):
[0143] FIG. 15 shows a flow diagram describing the initial clustering of documents. In step 1502, the local reverted index and dictionary of words/phrases is created. In one embodiment, InterMedia's reverted index is created on DOCUMENTS_TMP table. The table may not contain too many rows, because of performance reasons. In step 1504, the document summaries are generated and statistical data for the final clustering are prepared. Key algorithms used for this step may include 6 and 7. In embodiments, the 50 best words/phrases for each document are generated in WORDS_TMP table. In step 1506 global data is updated with local data, implementing key algorithm 3. This step is performed by updating TOKENS table with information collected in TOKENS_TMP and copy summaries from GISTS to DOCUMENTS. InterMedia indexes may also be created on DOCUMENTS and TOKENS tables. In step 1508, clusters hierarchy is generated by implementing key algorithm 4. The clusters hierarchy may be generated into T2 TOKENS table using an "is-substring" rule. In step 1510, cluster descriptions are generated by implementing key algorithm 5. The best five documents for each cluster are preferably generated into DOC2TOKEN table. In step 1512, elementary clusters with too many documents is found by implementing key algorithm 6. In this step, elementary clusters containing too many documents are found and this information is stored into LEAVES. Also, documents are assigned to elementary clusters and this information is saved into DOCS_FOR_LEAVES table. In step 1514, documents not covered by any elementary cluster are found by implementing key algorithm 6. In step 1516, the processing of FIG. 15 is completed.

Detail Description Paragraph (87):
[0145] Extract statistical information about words and their occurrences in documents

Detail Description Paragraph (93):
[0151] Document Tagging--assigns tags to the documents. The extracted tags are used later to generate XML stream for the document;

Detail Description Paragraph (94):
[0152] Dictionary Creation--document analysis and document tagging. The dictionary contains all words and phrases occurring in documents with their statistical information. It is updated during the analysis process and used later e.g., for document tagging;

Detail Description Paragraph (96):
[0154] From a logical point of view, the DC2 module 200B transforms unstructured textual data (web documents) into structured data (in form of tables in a relational data base). There are two sub-systems that keep interaction with the DC2 module, including the console 800 and Data Storage and Acquisition (DSA) module 100. In embodiments, the DC2 module performs its computation using two databases, the DSA database 100A and Dialog Control database 1600. The DC2 module obtains documents from the DSA database 100A and saves results to Dialog Control database 1600. The basic scenario of using the DC2 module includes, assuming that the System Administrator (CONSOLE) downloads a number of documents and wants to extract information about those documents:

Detail Description Paragraph (107):
[0165] In FIG. 17, the DC2 sub-modules are labeled "Extracting Information about Documents" and "Building Document Representation", and both are performed in the off-line mode. The "Extracting Information about Documents" 200A and "Building Document Representation" 200B provide information to both the document information at function block 1700 and the clustering of documents at function block 1702. The documents are then built into cluster hierarchies in function blocks 1704 and 1706. In the off-line mode, there is a dialog with the user at function block 1710 which communicates with the clustering hierarchy at function block 1706 and the document information at function block The user is able to retrieve this information at the user interface 1708.

Detail Description Paragraph (116):
[0174] In Document Tagging 1820, the dc2analysis assigns tags to the documents. The extracted tags are used later to generate XML stream for the document. A dictionary 1822 is also provided, which is a collection of all words occurring in the analyzed documents. The dictionary may be synchronically accessible.

Detail Description Paragraph (119):
[0177] FIG. 21 is a diagram outlining the document tagging of FIG. 18. In the General Tagging block 2102, there are three kinds of information included: keywords, summaries and language. In the Keyword Extraction block 2103, a list of five most significant words is generated for the document. In the language Recognition and Summarizing block 2104, the summary provides a gist of the document. The gist includes a couple of sentences taken from the document which reflect the subject of the document. The language provides information about the (main) language of the document. In the Special Tagging block 2106, the following tags may be generated, with other tags also contemplated for use with the present invention.

Detail Description Paragraph (125):
[0183] SUMMARY--the summary of document

Detail Description Paragraph (132):
[0189] splitting clusters into tree structure, so that, every leaf of that tree, called elementary cluster, has less number of documents than specified by a user

Detail Description Paragraph (140):
[0197] One of the main requirements for the matrix is to have the elementary clusters with less documents than a number specified by a user. A second requirement for the matrix is that the clusters should be labeled by single phrases or words which are good descriptions for the whole group of documents and should distinguish clusters appearing at the same level of cluster hierarchy.

Detail Description Paragraph (164):
[0221] The Skowron 2 taxonomy module 802 includes a completer module and a taxonomer module. The completer module is implemented in order to improve dialog quality by covering documents not belonging to any DC1 elementary clusters. In this way, the completer process creates additional clusters containing (directly or through subclusters) these uncovered documents. By way of example and referring to FIG. 25, InterMedia cluster for the phrase car 2500 is provided. This cluster may contain several InterMedia sub clusters, e.g., car rental 2500A, car dealers, 2500B. The problem is that many documents will contain the phrase car, but will not contain any longer phrase with car as subphrase.

Detail Description Paragraph (166):
[0223] FIG. 26 is a high-level view of the completer module external interactions. The completer (represented in function block 2604) runs after DC 1 analysis in function block 2602 and prior to the taxonomer (as discussed in more detail below). The completer module creates new clusters based on information provided by dc1 analysis. And, the taxonomer module splits large clusters both created by DC 1 analysis (InterMedia clusters) and by the taxonomer module (phrase clusters). Flow of documents is obtained from the database (function block 2600) and information flows between the taxonomy module and the database (represented at function block 2606).

Detail Description Paragraph (172):
[0229] FIG. 27 shows a flow diagram of the steps implementing the processes of the completer module. First, the data for the completer module is prepared by the DC 1 module 801. This data contains the list of clusters containing the uncovered documents, along with the list of all uncovered documents in each of these clusters. Having given clusters and uncovered documents inside them, the completer module, in step 2702, obtains the next processed DC1 non-elementary cluster. In step 2704, a determination is made as to whether any clusters are found. If no clusters are found, the process ends. If clusters are found, then for each given cluster C and the set of uncovered documents D inside this cluster a statistical sample is taken of the uncovered documents sample D in step 2706. In step 2708, a matrix is generated for the sample (using the matrix module). In step 2710, the words which best cover the same are found. In step 2712, new clusters are created corresponding to the best words found in step 2710. In other words, based on the matrix data, a set of phrases, Cover(sample(D))={w.sub.1, . . . , W.sub.n}, is found which becomes the names of new clusters. The new cluster labeled with the phrase w.sub.i will contain all the documents containing this phrase. The ideal situation is when every document contains at least one of the phrases w.sub.1, . . . , w.sub.n, and the number of documents containing the phrase w.sub.i (for each i) is not too large. This number multiplied by size(D)/size(sample(D)) should, in embodiments, not exceed the maximal size of the cluster that the taxonomer module is able to split. Finding of the set Cover(sample(D)) is performed according to greedy algorithm: first, the word covering the most of the documents is chosen, then the word covering the most documents that are not covered yet is chosen, and so on. The information on new clusters and the best documents may then be inserted into dialog tables. It should be noted that each subcluster added to the cluster by the completer module is processed once. Thus, if the new documents are fed into the engine and the completer process is run, the new subclusters are not added by completer module to the previously analyzed clusters.

Detail Description Paragraph (175):
[0232] For example, referring to FIG. 28, the DC1 elementary cluster for the phrase car rental 2600 has two parents: car 2600A and rental 2600B. Assume that the cluster contains too many documents and hence the problem is to decompose the cluster due to its size. The decomposition includes the creation of subtree of clusters with root in the car rental cluster. The leaves of the subtree satisfy (approximately) the completeness and decomposition conditions. The procedure creates at most two additional levels of clusters. These clusters include insurance 2602A, credits 2602B, wheel 2602C and Ford 2602D, as well as agents 2602A, and bank 2602A.sub.2. The first level could be indirect and elementary phrase clusters (indirect cluster: insurance and elementary clusters: agents, bank) and on the second only elementary clusters (credits, wheel, Ford).

Detail Description Paragraph (176):
[0233] The data for the taxonomer module is prepared by the DC1 module and the completer module. In fact the taxonomy module uses mainly size of document set of cluster for determining which of them have to be split. The taxonomer works according to the following general steps shown in flow diagram of FIG. 29. In step 2902, a schedule is generated for the elementary clusters produced by the DC Analysis. This

list of clusters should be greater than taxonomy.maxPhraseClusterSize property and the elementary DC1 clusters (i.e., tokens.is_elementary=1 in database). In step 2904, a schedule is generated for the phrase clusters produced by the completer module. This list should include clusters which size is greater than taxonomy.maxPhraseClusterSize property and phrase type (i.e., tokens.type=4 in database). In step 2906, each of the clusters are decomposed as discussed in more detail with reference to FIG. 30. In step 2908, the decomposition is finalized. This includes updating some of the global database structures.

Detail Description Paragraph (177):
[0234] FIG. 30 is a flow diagram showing the substeps of the decomposition step 2906 of FIG. 29. First, given cluster C and its phrase p:

Detail Description Paragraph (178):
[0235] Let docs(q1, . . . , qn) determine set of documents which contains within body all phrases q1, q2, . . . , qn.

Detail Description Paragraph (180):
[0237] Now, in step 3002, the sample of documents are taken from cluster C: sample(C). The statistical sample (which size is determined by taxonomy.sampleSize property) is taken if one of the following conditions is satisfied: (i) the number of documents in the cluster C (in docs(p)) is greater than taxonomy. bigCluster property or (ii) if the cluster C is elementary DC1 cluster and size of the set docs(p1, . . . , pn) is greater than taxonomy.bigCluster property. If so, then let alpha(C)=taxonomy.sampleSize/size(C); otherwise all documents in the cluster are taken (alpha=1). In step 3004, a matrix (words occurrence table for most relevant words) is generated for the sample. The matrix module is used to generate the matrix (referred to as child matrix). In step 3006, if the cluster C is elementary DC1 cluster and alpha=1, then the matrix for the documents form docs(p1, . . . , pn) is generated (referred to as parent matrix). Otherwise all references to the parent matrix are assumed to be equal to the child matrix. In step 3008, the set of alternatives which will be the base for the decomposition process is found based on the matrix data. The alternative is a pair <q, Q>, where q is a phrase and Q is a set of phrases. Note that set Q includes child matrix phrases q.sub.1, . . . , q.sub.k which have the following property: size of set of documents determined by phrases q and qi in the sample has size not grather than max(alpha(C) *taxonomy. max PhraseClusterSize, 10) and not less then max(alpha(C) *taxonomy.minPhraseClusterSize,1). The inequalities allow to satisfy approximately decomposition condition. The phrase q is taken from the parent matrix data, and q is not in Q. The alternatives are created sequentially. The general heuristics of this step includes:

Detail Description Paragraph (181):
[0238] 1. find the best phrase q from parent matrix data which has not been used in alternatives;

Detail Description Paragraph (183):
[0240] 3. if a phrase is used as q or in Q then global priority of the phrase is decremented.

Detail Description Paragraph (185):
[0242] Still referring to FIG. 30, in step 3010, based on the set of alternatives the decomposition subtree is found for the cluster. The general condition for this step is to find the smallest subet Alt={<q.sub.1, Q.sub.1>, . . . ,<qm, Qm>} of alternatives which gives the highest coverage coefficient which is determined as .vertline.(docs(Q1).orgate. . . . .orgate. docs(Qm)) .andgate. sample(C).vertline.. In step 3012, if the coverage coefficient does not satisfy the completeness condition then create additional alternatives called singletons. A singleton is a special kind of alternative in which set Q has exactly one element equal to phrase q. Searching conditions and the methods are the same as above. In step 3014, a decomposition tree is created such that for each created alternative 21 q, Q>:

Detail Description Paragraph (186):
[0243] 1. if Q has exactly one element s create child cluster with the phrase s and set type of the new cluster as elementary phrase;

Detail Description Paragraph (187):
[0244] 2. if Q has elements q.sub.1, . . . , q.sub.k create child cluster C' with the main phrase q and create its children C.sub.1, . . . , C.sub.k with the main phrases equal q.sub.1, . . . , q.sub.k respectively. Set type for C' equal indirect and for

C.sub.1, . . . , C.sub.k--elementary;

Detail Description Paragraph (189):
[0246] 4. set approximate size of the new clusters as .vertline.docs(p,q').vertline./alpha where q' is main phrase for the cluster. In step 3016, the information on the new clusters and the best documents in these clusters are inserted into dialog tables.

Detail Description Paragraph (192):
[0249] The Dialog Control module 300 allows the user's requests to be described as Boolean functions (called patterns) built from atomic formulas (words or phrases) where the variables are phrases of text. For example, a pattern may be represented as:

Detail Description Paragraph (195):
[0251] The patterns may be implemented, for example, by a set of five classes, including Pattern and subclasses Phrase, Or, And,, and Neg. The following code illustrates the use of these classes

Detail Description Paragraph (200):
[0256] (ii) It groups the retrieved documents into similarity clusters and returns to the user standard patterns of these groups. This step, by itself can, be defined as:

Detail Description Table CWU (1):
1 No. Process Algorithm Description 1 Analyzer Extract Ignore everything within SCRIPT plain text tags. Replace each tag with a single from HTML. space. Return the result. 2 Analyzer Extract Extract information from META, metain- TITLE and LINKS tags. formation from HTML. 3 Indexer & Generate Create reverted index of occurrences Statistics clusters. of words and phrases in documents. Phrases are taken from the knowledge base or recognized by primitive heuristics. Cluster is a set of documents containing the same word or phrase. Its label is defined as the word or phrase. 4 Indexer & Generate Build directed acyclic graph. Edge Statistics cluster (u,v) between phrases means, that u hierarchy. is the subphrase of v. 5 Indexer & Generate Count documents in each cluster. Statistics cluster Find the "most-representative-five" descriptions. documents for each cluster. 6 Indexer & Generate Find the best 50 (at most) words or Statistics statistical phrases for each document. data for additional clustering. 7 Indexer & Generate Extract a limited number of the most Statistics document representative sentences for the summaries. document.

Detail Description Table CWU (2):
2 Module Algorithm Description DC2 Document Summary provides a gist of the document: it Analyser summar- consists of a couple of sentences taken from the ization document which reflect the subject of the document. 1. Compute the basic weight of a sentence as a sum of weights of words in the sentence. This weight is normalized to some extent with respect to the length of a sentence, and very short and very long sentences are penalized. 2. Select sentences with highest weights and order them according to the order in which they occur in the input text. There are limits on summary lengths, both in number of characters and in number of sentences. DC2 Language Language recognition provides information Analyser recog- about the (main) language of the document; nition 1. Statistical models for each language are applied to the summary; 2. the model that models the text best (i.e., the one that "predicts" the text best) is assumed to reflect the actual language of the summary and, hence, of the whole input text. DC2 Com- Priority of word s is a sum of evaluating Analyser puting the measure of particular occurrence of s in the priority of document. The algorithm for computing the words in word priority is as follows: document First, we fix some constants: titleP = 31 (priority of words occurring in the title) headerP = 7 (priority of words occurring with header formats) empP = 3 (priority of words occurring with emphasis formats) Next, for every word s: s.priority = number of occurrences of s in the document; s.priority = s.priority + titleP*[number of occurrences of s in title] + (headerP + 5)*[number of occurrences of s with H1 tag] + (headerP + 4)*[number of occurrences of s with H2 tag] + (headerP + 3)* [number of occurrences of s with H3 tag] + (headerP + 2)*[number of occurrences of s with H4 tag] + (headerP + 1)*[number of occurrences of s with H5 tag] + (headerP)*[number of occurrences of s with H6 tag] + empP*[number of occurrences of s with some font format] DC2 Keyword Keywords are the most representative words for Analyser extraction a given document. Keywords are extracted as follows: 1. For each word s occurring in the document D compute the importance index for s using the formula: Importance(s,D) = = [Priority(s,D)/size(D)] log[N/DF(s)] 2. Select the 5 words with highest importance dc2 Admin- The main problem in parallel processing is analysis istration based on resource administration. In the and document analysis module, it is necessary to: control of .cndot. guarantee that every document stored in document

crawler data base is analyzed exactly one analysis time, processes .cndot. synchronize the access to the Dictionary which is a collection of all analyzed words. In dc2analysis the Control and Administration algorithm is based on the construction of 3 kinds of threads: Provider, Analyzer and Saver. .cndot. Provider downloads consecutive packages of documents from the crawler data base. .cndot. Analyzer gets documents from Provider, analyses them and sends results to Saver. .cndot. Saver saves results to disk. There is only one Provider and one Saver. The number of Analyzers may be depended on the number of terminals which are earmarked for computation.

Detail Description Table CWU (4):

```
4 void main() { Pattern *P = new Pattern(); Phrase fraza("Project"); char T[256]=""; P
= &(fraza * "House"); P = &(*P - "Construction"); printf(P->Pat2Text(T)); }
```

CLAIMS:

1. A method for analyzing and processing documents, comprising the steps of: building a dictionary based on keywords from an entire text of the documents, analyzing text of the documents for the keywords or a number of occurrences of the keywords and a context in which the keywords appear in the text; and clustering documents into groups of clusters based on information obtained in the analyzing step, wherein each cluster of the groups of clusters includes a set of documents containing a same word or phrase.

4. The method of claim 1, further comprising the step of splitting the groups of clusters into subclusters, the splitting step including: finding words which are representative for each of the group of clusters; generating a matrix containing information about occurrences of the top words in the documents from the groups of clusters; and creating new clusters based on the generating step which corresponds to the top words and a set of phrases.

5. The method of claim 1, wherein the analyzing step includes analyzing the documents for statistical information including word occurrences, identification of relationships between words, elimination of insignificant words and extraction of word semantics.

10. The method of claim 1, further comprising the step of summary generation of the documents, the summary generation being based on the assigned weights to the words and the appropriate weights of the sentences.

13. The method of claim 12, wherein the analyzing step includes applying linguistic analysis to the documents, the linguistic analysis being performed on one of titles, headlines and body of the text, and content including at least one of phrases and the words.

14. The method of claim 13, wherein the dictionary generates words that describe the contents of the documents, creates indexes for the documents, associates the documents with other documents to create concept hierarchy, clusters the documents using a tree-structure of the concept hierarchy and generates a best-suited phrase for cluster description.

18. The method of claim 1, wherein the clustering step is based on one of (i) a best-suited phrase or word from the documents and (ii) generation word conjunction templates for grouping the documents.

19. The method of claim 1, wherein the analyzing step includes extracting document meta information.

20. The method of claim 1, further comprising the steps of generating a cluster heirarchy for the groups of clusters; generting cluster descriptions, the clustering descriptions including words or phrases that generate a cluster of the groups of clusters and the number of the documents in the cluster; and assigning the documents to elementary clusters and indirect clusters.

22. The method of claim 1, further comprising the step of processing the documents, the processing including: creating reverted index of occurrences of words and phrases in the documents; building a directed acyclic graph; and extracting a limited number of representative sentences or words or phrases for the document.

24. The method of claim 23, wherein the clustering step includes the steps of: creating reverted index of occurrences of words and phrases in the documents; building a directed acyclic graph; and counting the documents in each group of clusters.

25. The method of claim 24, wherein the clustering step further includes: generating document summaries and statistical data for the groups of clusters; updating global data by using the document summaries; generating cluster descriptions of the groups of clusters by finding representative documents in the each cluster of the groups of clusters; finding elementary clusters associated with the groups of clusters which contain more than a predetermined size of the documents; and storing the elementary clusters in storage.

27. The method of claim 1, wherein the analyzing step includes the steps of: computing a basic weight of a sentence as a sum of weights of the words in the sentence; normalizing the weight with respect to a length of the sentence; selecting sentences with highest weights; ordering the sentences with the highest weights in an order which they occur in the input text; providing a priority to the words by evaluating a measure of particular occurrence of the words in the documents; and extracting the keywords from the documents which are representative for a given document, the keywords being extracted as follows: for each word s occurring in the document D compute an importance index for s using the formula: 1 Importance (s , D) = = [Priority (s , D) / size (D)] log [N / DF (s)] where N is a number of all the documents and DF(s) is the number of all the documents which contain the word s.

32. A system for analyzing and processing documents, comprising the steps of: a module for building a dictionary based on the keywords from an entire text of the documents, a module for analyzing text of the documents for the keywords or a number of occurrences of the keywords and a context in which the keywords appear in the text; and a module for clustering documents into groups of clusters based on information obtained in the analyzing step, wherein each cluster of the group of clusters is a set of documents containing a same word or phrase.

33. A machine readable medium containing code for analyzing and processing documents, comprising the steps of: building a dictionary based on the keywords from an entire text of the documents, analyzing text of the documents for the keywords or a number of occurrences of the keywords and a context in which the keywords appear in the text; and clustering documents into groups of clusters based on information obtained in the analyzing step, wherein each cluster of the group of clusters is a set of documents containing a same word or phrase.

WEST

□ Generate Collection Print

L18: Entry 4 of 5 File: PGPB Dec 12, 2002

DOCUMENT-IDENTIFIER: US 20020188587 A1
TITLE: System, method and apparatus for generating phrases from a database

Summary of Invention Paragraph (4):
[0002] The vast amount of text and other types of information available in electronic form have contributed substantially to an "information glut." In response, researchers are creating a variety of methods to address the need to efficiently access electronically stored information. Current methods are typically based on finding and exploiting patterns in collections of text. Variations among the methods and the factions are primarily due to varying allegiances to linguistics, quantitative analysis, representations of domain expertise, and the practical demands of the applications. Typical applications involve finding items of interest from large collections of text, having appropriate items routed to the correct people, and condensing the contents of many documents into a summary form.

Brief Description of Drawings Paragraph (23):
[0036] FIG. 20 shows an overview of one embodiment of the phrase extraction process;

Brief Description of Drawings Paragraph (28):
[0041] FIG. 21 illustrates one embodiment of culling the extracted phrases;

Detail Description Paragraph (5):
[0050] For one embodiment, the contextual associations of a term provide contextual meaning of the term. For example, the term "fatigue" can refer to human physical tiredness such as "Fatigue impaired the person's judgment." Or "fatigue" can refer to breakdown of the structure of a material such as "Metal fatigue caused the aluminum coupling to break." A first aggregation of associations between term pairs such as: "fatigue" and "person", "fatigue" and "impaired", and "fatigue" and "judgment" can be clearly differentiated from a second aggregation of associations such as "metal" and "fatigue", "fatigue" and "aluminum", "fatigue" and "coupling", and "fatigue" and "break". Thus, when searching a database of subsets for subsets containing the notion of "fatigue" in the sense of human physical tiredness, subsets having greater similarity to the first aggregation of associations are more likely to include the appropriate sense of "fatigue", so these subsets would be retrieved. Further, the contextual associations found in the retrieved subsets can both refine and extend the contextual meaning of the term "fatigue".

Detail Description Paragraph (8):
[0053] Relevance ranking a collection of models is a method of quantifying the degree of similarity of a first model (i.e., a criterion model) and each one of the models in the collection, and assigning a rank ordering to the models in the collection according to their degree of similarity to the first model. The same rank ordering can also be assigned, for example, to the collection of identifiers of the models in the collection, or a collection of subsets of a database represented by the models of the collection. The features of the criterion model are compared to the features of each one of the collection of other models. As will be described in more detail below, the features can include the relations and the contextual measurements, i.e. the relational metric values of the relations in the models. The collection of other models is then ranked in order of similarity to the criterion model. As an example: the criterion model is a model of a query. The criterion model is then compared to a number of models of narratives. Then each one of the corresponding narratives is ranked according to the corresponding level of similarity of that narrative's corresponding model to the criterion model. As another alternative, the criteria model can represent any level of text and combination of text, or data from the database, or combination of segments of sets of databases.

Detail Description Paragraph (287):
[0323] FIG. 19 shows one embodiment of an overview of the phrase discovery process 1900. The phrase discovery process is described in more detail below. First, a relevant text is provided in block 1902. The provided relevant text can be any text that contains the topic of interest, and preferably text that prominently contains the topic of interest. For example, if the topic of interest is "aircrew fatigue", then aircrew fatigue should be among the prominent topics in the provided relevant text. The relevant text can be any quantity of text such as a passage, a paragraph, a narrative, a collection of narratives, or larger selections of text. Phrases are extracted from the provided relevant text in block 1904. The extracted phrases can include all phrases that occur in the relevant text. Alternatively, the extracted phrases can include a selected number of the phrases that occur in the relevant text. The extracted phrases are culled in block 1906. The culled phrases are then input to a gathering process in block 1908. The gathering process gathers phrases that are contextually associated, that is, phrases that are prominent in the local context of the provided relevant text, but are not prominent in the global context of a larger collection of similar text. The phrases resulting from the gathering process 1908 are output in block 1910.

Detail Description Paragraph (289):
[0325] Phrase extraction is a process of identifying and collecting a number of sequences of terms that occur within a larger sequence of terms contained in one or more subsets of a database. One embodiment of phrase extraction obtains phrases from a collection of text. Phrase extraction can identify phrases that occur one or more times in the input sequence of terms without reference to any pre-existing lists of phrases, and without recognition of the grammatical structure of language. Phrase extraction uses each term in the input sequence of terms as a first term in a number of phrases. First, a phrase consisting of a single (1) term is identified. Then, starting with the single term, a phrase of two (2) terms is identified. Processing continues until phrases containing any number of terms, up to a selected number (N) of terms, are identified. Then, a subsequent term is identified in the sequence of terms, and another set of phrases of length 1 to N are identified. The process continues until every term in the input sequence of terms has been used as a starting term for a set of phrases of length 1 to N. In one alternative, a count of the unique phrases is maintained and only one copy of each unique phrase is output along with the corresponding frequency of the unique phrase.

Detail Description Paragraph (290):
[0326] In one alternative, phrase extraction can include one or more sets or classes of special terms to determine whether and to what extent a term from one set of special term's is allowed to appear in a particular position within a phrase. Based on the terms membership in the set of special terms and the term's presence in the phrase, the phrase may or may not be identified as an acceptable phrase. Only acceptable phrases are then output to the culling process. In one alternative, the special terms include one or more sets of stopterms. In one alternative, the special terms include one or more sets of stopterms. In one alternative, a set of stopterms includes zero or more terms that occur in the relevant text. In another alternative, a set of stopterms can include conventional stopwords such as articles and conjunctions. Stopterms can also include punctuation.

Detail Description Paragraph (291):
[0327] The culling process reduces the number of extracted phrases. In one embodiment, the culling process eliminates a phrase that only occurs as part of another, longer phrase within the provided relevant text from which the phrases were obtained. In one alternative, the previously extracted phrases can be input to the culling process. The phrases input to the culling process are collected in a list of candidate phrases. A first phrase from the candidate phrases is selected and the selected phrase is then examined to see if the selected phrase is contained within any of the other candidate phrases in the candidate phrase list. If the selected phrase is contained in another candidate phrase (i.e. a containing phrase) in the candidate phrase list, then the frequencies of the selected phrase and the containing phrase are examined. And if the frequency of the selected phrase is not greater than the frequency of the containing phrase, then the selected phrase only occurs in the provided relevant text as part of the containing phrase. Therefore, the selected phrase is not a stand-alone phrase and is therefore deleted. Each of the phrases in the candidate phrase list are tested as described above. The candidate phrases that remain in the candidate phrase list after the culling process is complete are then output. In one alternative, the phrases are output to a gathering phrases process.

Detail Description Paragraph (293):

[0329] The gathering phrases process can also be an iterative process. When the gathering phrases process is iterative, each iteration after the first gathering of phrases includes a phrase search where the previously gathered phrases as the input query. The output of the phrase search includes a new body of provided relevant text, from which additional phrases are obtained, as described below. Thus, the iterative process uses feedback of associated phrases to obtain additional contextually associated phrases. The database searched by the phrase search can include the larger collection of similar text, and alternatively, an additional collection of text. The iterative gathering process can also include a process of extracting additional phrases from the new body of provided relevant text, and can also include a culling process to reduce the number of extracted phrases, to produce additional phrases that are contextually associated. The additional phrases can be sorted according to the corresponding degrees of contextual association and combined in sorted order with previously gathered phrases.

Detail Description Paragraph (295):
[0331] FIGS. 20-20E illustrate various embodiments of the phrase extraction process 1904. FIG. 20 shows an overview of one embodiment of the phrase extraction process 1904. First the phrase starting positions are processed within the relevant text in block 2002. The phrase starting positions include the terms in the relevant text that the process will use to begin each iteration of the phrase extraction process. In one alternative, a number of selected starting position terms are extracted as a number of single-term phrases. Selected multi-term phrases are extracted in block 2004. Multi-term phrases include two or more terms. The first term of each multi-term phrase is one of the phrase starting position terms. The resulting phrase list is output to the next sub-process in block 2006.

Detail Description Paragraph (296):
[0332] FIG. 20A illustrates one embodiment of the phrase starting positions process 2002. A first term in the relevant text is identified in block 2010. The first term is then identified as both T1 and T2 in block 2011. Next, if there is a term subsequent to T1, then T1 is not the last term in the relevant text and it is possible that T1 is an acceptable first term in a multi-term phrase, therefore determine if T1 is a stopterm in block 2013, or alternatively, if T1 is a starting stopterm in block 2013A. If T1 is a stopterm in 2013, or if T1 is a starting stopterm in 2013A, then T1 is not an acceptable first term in a multi-term phrase, and therefore identify the term subsequent to T1 as both T1 and T2 in block 2014. The process continues at block 2012. If T1 is not a stopterm in block 2013, or alternatively, if T1 is not a starting stopterm in 2013A, then T1 is an acceptable first term in a multi-term phrase and a potentially acceptable single term phrase, therefore T1 is saved in the phrase list (PL) as a single term phrase in block 2015 according to the subprocess shown in FIG. 20B, as described below. Next, selected multi-term phrases are extracted at the starting position T1 in block 2004 according to the process described in FIG. 20D or FIG. 20E, as described below. After extracting phrases in block 2004, the phrase extraction process begins at a new starting position by continuing the process at block 2014.

Detail Description Paragraph (302):
[0338] Having a single class of stopterms, combined with determination of the position of stopterms within a phrase, may be sufficient for some applications of the phrase extraction process, but having additional classes of terms provides additional control and refinements in extracting phrases having particular forms. A process using multiple classes of terms is illustrated in FIG. 20E, described below. FIG. 20E illustrates an alternative embodiment of extracting selected multi-term phrases at each starting position in the text. The process of FIG. 20E differs from the process of FIG. 20D in that the process illustrated in FIG. 20E includes use of a number of classes of stopterms and a class of interior-only terms. Three classes of stopterms are illustrated: starting stoptersm, interior stopterms, and ending stopterms. A starting stopterm is a term that may not be the first term of a phrase. An interior stopterm is an interior term that may appear only up to a pre-selected number of times in a phrase (including zero times). An ending stopterm is a term that may not be the last term of a phrase. When distinguishing among the three classes is unnecessary, a stopterm in any class is merely referred to as a stopterm. An interior-only term is a term that is not an interior stopterm and may not be the first or last term of a phrase.

Detail Description Paragraph (304):
[0340] In another application, the phrase extraction process can be used for highly targeted phrase extractions, such as finding certain prepositional phrases. In one alternative, highly targeted extractions can be done by defining all vocabulary words

except prepositions as starting stopterms, using a conventional stoplist for the ending and interior stopterms, and allowing up to two interior stopterms. Such phrases as "on board", "in the cockpit", "at altitude", and "below the other aircraft", would be accepted, while all phrases not starting with a preposition would be rejected. Interior-only terms could be used to further limit the acceptable phrases. Additional general classes of terms, such as ending-only terms, can also be envisioned.

Detail Description Paragraph (306):
[0342] FIG. 21 illustrates one embodiment of culling the extracted phrases in block 1906 of FIG. 19. The first phrase from the candidate phrase list (CPL) is identified as P1 in block 2102. Several phrases from the CPL are identified. Each one of the identified phrases includes P1 as a proper subset in block 2104 i.e. P1 is only a portion of each one of the phrases. A first one of the phrases is identified as P2 in block 2106. If the frequency of P1 is equal to the frequency of P2 in block 2108 then P1 is eliminated from the CPL in block 2110 and the process continues at block 2116 below. If the frequency of P1 is not equal to the frequency of P2 in block 2108, then a phrase subsequent to P2 is selected as P2 in blocks 2112, 2114 and the new P2 is input to block 2108 above. If there are no more phrases subsequent to P2 in block 2112, then a phrase subsequent to P1 in the CPL is selected as P1 2116, 2118 and the subsequent P1 is processed beginning with block 2104. If there are no more phrases subsequent to P1 in the CPL then the phrases in the CPL are output to the process of gathering related phrases in block 1908 of FIG. 19.

Detail Description Paragraph (310):
[0346] The phrase search in block 2214 outputs a ranked list of subsets from the database and a selected number of the ranked list of subsets are then designated as the relevant text and input to the extract phrases process described in FIG. 20 in block 1904. The phrases extracted from the extract phrases process in block 1904 are then input to the process of culling the extracted phrases described in FIG. 21 in block 1906. The phrases output from the process of culling the extracted phrases in block 1906 are then ranked at block 2204 and the process repeats, until the number in the phrase search counter is greater than the pre-selected number of phrase searches.

Detail Description Paragraph (311):
[0347] FIG. 22A illustrates one embodiment of ranking the phrases output from the extracting and culling processes of block 2204 of FIG. 22. First, the relevant text from which the phrases were processed is selected in block 2224. A local model is then created in block 2226. A local model is a contextual model of subsets of the provided relevant text from which the phrases were extracted and culled. All of the relevant text could be modeled in one embodiment. Alternatively, only a selected number of subsets of the provided relevant text that are also the most representative of the provided text are also modeled. One embodiment of a local model includes isolating distinct subsets from one another within the selected relevant text. Another embodiment of a local model includes inserting several non-term "buffer terms" between distinct subsets. A non-term buffer term includes a set of text designated as space filler. Another embodiment of a local model includes generating a vocabulary list that includes the terms that occur in the selected relevant text and the frequency of each term.

Detail Description Paragraph (312):
[0348] Next, a global model is selected in block 2228. A global model can include a contextual model of the entire database or a single relational model of a number of subsets. A global model can also include a single relational model of a number of subsets wherein the number of subsets is greater than the number of subsets used to generate the local model. Alternatively, a global model can include a single relational model of a number of subsets wherein the subsets include the relevant text from which the selected phrases were extracted and culled. A global model can also include a single relational model of subsets wherein the subsets include text that is similar to the relevant text from which the selected phrases were extracted and culled. A global model can also include a number of relational models wherein each model represents one subset. A global model can also include creating a single relational model of a number of subsets by reducing the relations to unique relations. This process is similar to reducing the relations in a query described in keyterm search above, except reducing relations from all of the subset models, not just the subset relations matching a query. For another alternative embodiment a global model also includes limiting unique global model relations to only those relations having the same term pairs as relations in the local model.

Detail Description Paragraph (316):
[0352] FIG. 22D illustrates one embodiment of emphasizing the locally relevant phrases

and de-emphasizing the globally relevant phrases in block 2238 of FIG. 22B. First the re-weighted model is selected in block 2260 and the processed phrases are selected in block 2262. Alternatively, a weight could also be determined for each one of the processed phrases. The weight for each one of the processed phrases could also be set to a pre-selected value such as 1. A frequency of occurrence of the phrase within the selected relevant text could also be determined and used as the phrase weight. The selected phrases are then compared to the re-weighted model in block 2264. The selected phrases are then ranked in order of relevance to the re-weighted model in block 2266. The comparison in block 2264 can be a process similar to the comparison process in keyterm search described in FIG. 10 above. Thus, each phrase is modeled as a subset of the database, and the re-weighted model is used as a criterion model. The criterion model (that is, the re-weighted model) is compared with the subset models which represent the phrases to determine the degree of similarity of the criterion model and each of the phrase models. In addition, the ranking of the phrases in block 2266 can be done using the process of ranking subsets in keyterm search described above. Thus, the phrases are ranked on their degree of similarity to the re-weighted model.

Detail Description Paragraph (320):
[0356] The following Table 3.1 shows 50 of 420 phrases related to the topic of fatigue. The 420 phrases were extracted from three sets of 200 narratives that were found to be most relevant to the topic of fatigue. The frequency of each phrase within a set of 200 narratives is shown in the first column. This list shows, for example, that in the context of fatigue, "rest period(s)", "reduced rest", and "crew rest" are the most prominent concerns. Further, these are greater concerns than "continuous duty", "duty period", and "crew duty". The list also shows that "crew scheduling" ranks high among the concerns of the reporters in the context of fatigue. Other prominent concerns include: "reserve or standby", "rest requirements", "crew fatigue", "continuous duty overnight(s)", "adequate rest", "minimum rest", "required rest", "plt fatigue" (i.e., pilot fatigue), and "compensatory rest". The prominence of these fatigue-related phrases parallels the prominence of these concerns in the industry.

Detail Description Paragraph (339):
[0374] The above described methods and processes of keyterm search, phrase search, phrase generation and phrase discovery have been described and illustrated in terms of information retrieval (IR) embodiments. In IR: terms are symbols or elements of a data set, subsets are collections of symbols, databases are collections of subsets, each relation is binary and links a symbol pair, and quantification of relations is based on contextual associations of symbols within subsets. Further, models are collections of symbol relations, the models can be aggregated, the models can represent subsets, databases, and queries, models can be ranked on similarity to other models, and sequentially grouped terms are derived from models and subsets.

Detail Description Paragraph (341):
[0376] As with term pair relations in the IR embodiment, quantification of entity pair relations in the real world can also be based on contextual associations. In the real world, the scope of that context is space, time, causality, and thought. Thus, the notion of context is not limited to proximity relations among symbols within a subset. Instead, real-world context is a much broader concept, one that is more fully represented by the term "metonymy" in the sense developed by Roman Jakobson (Jakobson, R.: "Two aspects of language and two types of aphasic disturbances" (1956), (pp. 95-114) and "Marginal notes on the prose of the poet Pasternak" (1935), (pp. 301-317), in K. Pomorska and S. Rudy (Eds.), Language in Literature. Belknap Press, Cambridge, Mass., 1987). Jakobson asserted that the interpretation of a symbol or entity is derived from both its similarity to others and its contextual association with others. Thus, the contextual meaning of a symbol or entity is determined by its connections with others in the same context, that is, by its metonymic relations with others. This notion of metonymy, of contextual meaning, is a fundamental structural component of narrative text, symbol systems, and human behavior, according to Jakobson.

WEST

End of Result Set

☑ Generate Collection Print

L18: Entry 5 of 5 File: PGPB Nov 21, 2002

DOCUMENT-IDENTIFIER: US 20020174101 A1
TITLE: Document retrieval system

Summary of Invention Paragraph (7):
[0007] The resulting document signature is viewed as a vector of terms with associated weights, and as such occupies a multi-dimensional space within the features of all documents in the collection. As queries and documents may both be prepared and represented in this way, it was found that it was possible to measure the similarity between queries and documents trigonometrically, using vector-space analysis [Salton and McGill, 1983]. Under this scheme, the query vector is compared with each document vector in the collection using the formula: 1 Similarity (Q i , D j) = k = 1 t (w jk w ik) k = 1 t (w ik 2) .times. k = 1 t (w jk 2)

Summary of Invention Paragraph (8):
[0008] Where Q.sub.i is a query vector comprising a set of weights w.sub.ik and D.sub.j is a document vector comprising, a set of weights w.sub.jk. The formula is a `cosine` vector similarity measure, and provides the cosine angle between the query vector and the document vector.

Summary of Invention Paragraph (9):
[0009] For each document-query comparison, a score is produced representing the similarity or relevance of the document to the query. During the retrieval process, documents are retrieved and presented in descending order of relevance to the query.

Summary of Invention Paragraph (38):
[0038] In tandem with the system according to the first aspect of the invention, it is advantageous to provide a method of providing concise summaries of documents to facilitate use of the system.

Detail Description Paragraph (9):
[0062] Given a query Q.sub.j=(w.sub.j1,w.sub.j2, . . . ,w.sub.jt) and a document D.sub.i=(w.sub.i1,w.sub.i2, . . . w.sub.it), where w.sub.j and w.sub.i are the weights of the query keywords and document keywords respectively, the similarity is given by: 2 Similarity (Q i , D i) = k = 1 t w jk w ik

Detail Description Paragraph (11):
[0064] Rather than providing a relevance measurement based solely upon a `bag of words` (i.e. a set of keywords in any order), the system illustrated in FIG. 2 measures the relevance of documents on the basis of similarities between phrases representing queries and phrases representing documents.

Detail Description Paragraph (20):
[0073] Generally, known vector-space analysis methods and document similarity measurement methods, normalise the weights of keywords by using the following formula which produces vectors in which the sum of the keyword weights =1.0: 4 w i = w i k = 1 t (w ik 2)

Detail Description Paragraph (28):
[0081] Given a query Q.sub.j=(w.sub.j1,p.sub.j1,w.sub.j2p.sub.j2, . . . , w.sub.jt,p.sub.jt) and document D.sub.i=(w.sub.i1,p.sub.i1,w.sub.i2p.sub.- i2, . . . , w.sub.it,p.sub.it), where W.sub.j (and p.sub.j) and w.sub.i (and p.sub.i) are the weights (and positions) of the query and document keywords respectively the similarity is given by: 5 Similarity (Q i , D i) = k = 1 t w jk w ik + k = 1 t (p jk p ik)

Detail Description Paragraph (42):
[0095] By extending the present system to support multiple users, the system is able to unite users with similar interests and, by presenting the differences between these similar `interests`, to demonstrate to them subtly different approaches of keyword usage, as well as providing the results of previous searches. This will alert users to the presence of certain keywords they otherwise might not know about. It is important, however, to prevent too many similar interests from being shared, as this could overwhelm the user. The system therefore only shares interests if the level of similarity between the interests falls between certain (user selectable) bounds. This level of similarity is calculated in the same manner as that between documents and queries.

Detail Description Paragraph (43):
[0096] The `interest sharing` process is carried out in two ways. Firstly, pre-search collaboration is used. During query formulation, the system attempts to retrieve a user's interests based on the keywords they are entering (in the same manner as document retrieval). If it is unable to do this (for example, because the user currently has no relevant interests), the system attempts to trigger spheres of interest in other users'profiles, sorting the results by similarity in order to obtain the best possible match for the user. Furthermore, the interests returned are compared with the assistant's existing interests and may be retained for future use if they are deemed similar enough. This approach allows the system to `bootstrap` itself in order to start providing a service more quickly.

Detail Description Paragraph (44):
[0097] The second way in which the `interest sharing` process is carried out is via post-search collaboration. Whilst pre-search collaboration provides `emergency help` for a user, post-search collaboration provides a mechanism for a more generalised learning enhancement. Under this approach, whenever the system is idle, it will attempt to augment each user's profile with interest nodes from other users'profiles. This is carried out by using each interest node in a user's profile to trigger similar interests in other profiles. If the similarity between a user's interest node and those triggered in other profiles falls within `sharing constraints` defined by the user, then it will be added to that user's profile, together with information such as the other user's email address to facilitate personal contact, as well as direct links to the documents found useful by the other user. This form of collaboration is intended to provide the opportunity to unite similar users, present ideas for `different` searches and to determine whether the search proposed by a user has already been carried out by another user (by offering the results of previous searches).

Detail Description Paragraph (52):
[0105] 3. Clustering of similar phrases within the document. Following standard methods of extraction-based summarisation [Salton & Singhal, The automatic Text Theme Generation and the Analysis of Text Structure, Cornell University Technical Report TR 94-1438, 1994], all phrases extracted from the document are clustered into sets of similar phrases. Within this approach this is achieved by using each phrase to trigger every other phrase within the document. Thus each phrase will produce a variably sized set of `similar` phrases. The largest of these sets is taken to be an indication of the `average content` of the document. The final stage in producing the summary is to sort these phrases into their original order within the document.

Detail Description Paragraph (54):
[0107] Variables that may be used to affect the above described method include varying the trigger threshold of the phrase neurons to produce differently sized summary phrase sets, influencing the phrases contained in the phrase sets by centring the clustering around a `centre phrase`. This could be used to pick out important points from documents when indexing within a domain-specific context. For example if the system were indexing curricula vitae, a centre phrase of `research interests hobbies include` would force the indexing of phrases connected with document creator's research interests and hobbies. A further variable comprises introducing an upper threshold to similarity above which neurons will not fire. This would enable wider coverage of the clustering process by avoiding inclusion of very similar or repeated phrases and hence phrase duplication and redundancy.